

## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

SGT

CR151686

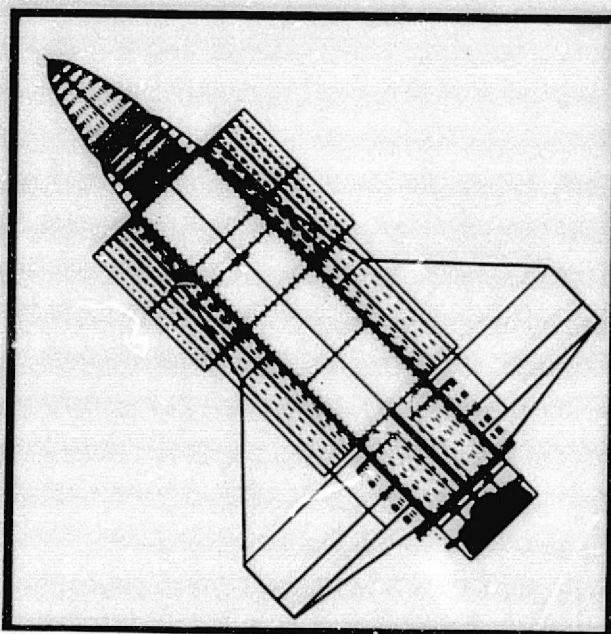
Programmer's

Manual

June 1977

# Thermal Radiation Analysis System

## TRASYS II



(NASA-CR-151686) THERMAL RADIATION ANALYSIS  
SYSTEM (TRASYS 2), PROGRAMMER'S MANUAL  
(Martin Marietta Corp.) 300 p HC A13/MF A01

N78-21792

CSCL 09B

Unclas

G3/61 14713



**MARTIN MARIETTA**

MCR-74-107 (Revision 1)  
Contract NAS9-14318  
DRL Number T-815  
DRL Line Item 7  
DRL Number DM-055TB

Programmer's  
Manual

June 1977

---

THERMAL RADIATION  
ANALYSIS SYSTEM

T  
R  
A  
S  
Y  
S  
I

Authors:

Robert J. Connor  
Ronald E. Paulson  
Richard G. Goble  
Carl L. Jensen

Approved



Carl L. Jensen  
Program Manager

MARTIN MARIETTA CORPORATION  
P.O. Box 179  
Denver, Colorado 80201

## FOREWORD

---

The Martin Marietta Thermal Radiation Analyzer System (TRASYS) program marks the first instance that thermal radiation analysis has been put on the same basis as thermal analysis using program systems such as MITAS and SINDA. As with these thermal analyzer programs, the user is provided the powerful options of writing his own executive or driver logic and choosing, among several available options, the most desirable solution techniques for the problem at hand. In addition, TRASYS provides many features never before available in a single radiation analysis program. Among the more important are:

- A 1000-node problem size capability with shadowing by intervening opaque or semitransparent surfaces;
- A choice of diffuse, specular, or diffuse/specular radiant interchange solutions;
- A capability for time-variant geometry in orbit;
- A choice of analytically determined or externally supplied shadow data for environmental flux calculations;
- Form factors and environmental fluxes computed using an internally-optimized number of surface grid elements, selected on the basis of user-supplied accuracy criteria;
- A general editing capability for updating thermal radiation model data stored on tape;
- A plot package that provides a pictorial representation of the user's geometry.

TRASYS is indebted to a number of predecessor programs in the thermal radiation analysis field. The major contributors were HEATRAT, MTRAP version 2.0, RADFAC, and the MRI computer program for determining external radiation absorbed by the Apollo spacecraft.

This programmers' manual represents an effort to provide scientific programming personnel with the descriptive material necessary to reach an understanding of the various program segments. Due to the highly modularized design of TRASYS, there are 166 preprocessor subroutines and over 300 processor library subroutines described herein. Although this is a rather large number in total, the individual subroutines have a more moderate size, so that the user can develop a working understanding of any routine after devoting a reasonable amount of effort to reading the Fortran code and consulting the appropriate material herein.



This manual was generated for the National Aeronautics and Space Administration's Lyndon B. Johnson Spacecraft Center under NASA Contracts NAS9-13033 and NAS9-14318, Development of a Thermal Radiation Analysis/Heat Rate Computer Program System. The technical monitoring for this program was provided by Mr. Robert A. Vogt of the Thermal Technology Branch of the Structures and Mechanics Division, NAS Lyndon B. Johnson Spacecraft Center. His helpful suggestions during the development of TRASYS are gratefully acknowledged.

TRASYS would not exist without the superb design and programming efforts of Messrs. Ronald E. Paulson and Robert J. Connor, who were responsible for generating the majority of the TRASYS code. Their efforts are gratefully acknowledged. Extensive thanks are also due to Mr. G. M. Holmstead for his efforts in developing the direct irradiation program segment and for the valuable consulting effort he performed during the course of program development. Mr. Richard G. Goble is also recognized for his praiseworthy efforts in developing the specular-diffuse radiation interchange segment, the orbit plotter segment, and for his solutions of many knotty problems that cropped up during program checkout.

## REVISION SCHEDULE

Revision 1

June 1977

## CONTENTS

---

	<u>Page</u>
I. INTRODUCTION . . . . .	I-1
A. What Is TRASYS? . . . . .	I-1
B. System Structure . . . . .	I-2 and I-3
II. PART 1 - PREPROCESSOR . . . . .	II-1
A. Segment Definitions . . . . .	II-1
B. Subroutine and Function Descriptions - Preprocessor . . . . .	II-9
C. File Definitions - Preprocessor . . . . .	II-85
D. Variable Definitions - Preprocessor . . . . .	II-91
E. Entry Points - Preprocessor . . . . .	II-107
III. PART 2 - PROCESSOR LIBRARY . . . . .	III-1
A. Segment Definitions . . . . .	III-1
B. Subroutine and Function Descriptions - Processor Library . . . . .	III-19
C. File Definitions - Processor Library . . . . .	III-145
D. Variable Definitions - Processor Library . . . . .	III-155
E. Entry Points - Processor Library . . . . .	III-191

## Figure

---

I-1 Basic Flow in Using an Applications Program . . . . .	I-2
I-2 Basic Flow in Using TRASYS . . . . .	I-2
I-3 Internal Flow of TRASYS . . . . .	I-3
II-1 Diagram of Preprocessor Segment Structure . . . . .	II-2
III-1 Segment AQCAL Flow Diagram . . . . .	III-2
III-2 Segment DICAL Flow Diagram . . . . .	III-3
III-3 Segment DRCAL Flow Diagram . . . . .	III-4
III-4 Segment FFCAL Flow Diagram . . . . .	III-6
III-5 Segment RBCAL Flow Diagram . . . . .	III-7
III-6 Segment GBCAL Flow Diagram . . . . .	III-9
III-7 Segment NPLOT Flow Diagram . . . . .	III-10
III-8 Segment OPLOT Flow Diagram . . . . .	III-11
III-9 Segment PLOT Flow Diagram . . . . .	III-12
III-10 Segment QOCAL Flow Diagram . . . . .	III-14
III-11 Segment RCCAL Flow Diagram . . . . .	III-15
III-12 Segment RKCAL Flow Diagram . . . . .	III-16
III-13 Segment SFCAL Flow Diagram . . . . .	III-17

## I. INTRODUCTION

---

### A. WHAT IS TRASYS?

TRASYS, the Martin Marietta Thermal Radiation Analysis System, is a digital computer software system with a generalized capability to solve the radiation-related aspects of thermal analysis problems. When used in conjunction with a generalized thermal analysis program such as the Systems Improved Numerical Differencing Analyzer (SINDA) program, any thermal problem that can be expressed in terms of a lumped-parameter, radiation-conductor thermal network can be solved.

The function of TRASYS is twofold. It provides:

- 1) Internode radiation interchange data;
- 2) Incident and absorbed heat rate data from environmental radiant heat sources.

Data of both types are provided in a format directly usable by the thermal analyzer programs.

A primary feature of TRASYS is that it allows the user to write his own executive or driver program, which organizes and directs the program library routines to solve each specific problem in the most expeditious manner. The user also may write his own output routines; thus, the data output can directly interface with any thermal analyzer using the R-C network concept.

Other outstanding features of TRASYS include:

- 1) A 1000-node allowable problem size;
- 2) The ability to accommodate time-variable problem geometry;
- 3) An editing capability that allows the combination or separation of multiple thermal radiation models;
- 4) A plot package that provides pictorial plots of input geometry and orbit data, as well as output data.

The TRASYS system consists of two major components: the preprocessor, and the processor library. The preprocessor has two major functions. First, it reads and converts the user's geometry input data into the form used by the processor library routines. Second, it accepts the user's driving logic (written in the TRASYS-modified FORTRAN language) that directs user-provided and/or

library routines in the solution of the problem. The processor library consists of FORTRAN language routines that perform the functions commonly needed by the user. The user has, in some cases, a choice of solution techniques for performing the same function.

## B. SYSTEM STRUCTURE

In the usual engineering environment, a programmer is commissioned to prepare an applications program that is subsequently made available to the engineer on a production basis. The engineer supplies input data and receives output data, as shown in Figure I-1.

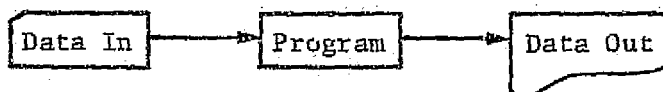


Figure I-1  
*Basic Flow in Using an Applications Program*

In most cases, changes to the logic and equations are difficult for the program user to implement conveniently since they must be written in a computer-oriented language and may be submitted through a formal programming organization. When TRASYS is used, however, the engineer need only call on the programmer to supply a standard deck of computer-oriented "control cards" that will call the various elements of the system into action in the proper sequence. The engineer then formulates his problem in the engineering-oriented TRASYS language, assembling both data and solution techniques (i.e., logic and equations) into this card deck, which then serves as the complete input to the TRASYS system. Programmer support has been minimized since the bulk of the programming effort is already built into the TRASYS pre-processor and processor library. The engineering user need only specify the data and the order and type of "program building blocks" he deems necessary to solve his problem (see Fig. I-2).

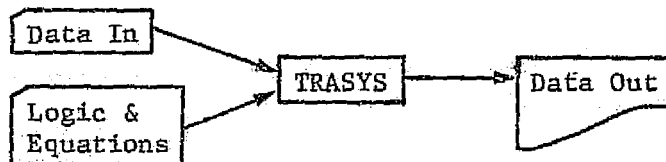


Figure I-2 *Basic Flow in Using TRASYS*

It should be evident that TRASYS is much more than an applications program. It has, in fact, all the functions and capabilities of a special-purpose operating system. Since most computers currently used in engineering environments already have operating systems built around a FORTRAN compiler, TRASYS is designed to augment the

existing FORTRAN system. Hence, the TRASYS library serves as an extension to the existing FORTRAN library, and the TRASYS program serves as a preprocessor to (i.e., it precedes) the existing FORTRAN compiler. This augmentation arrangement is illustrated in Figure I-3.

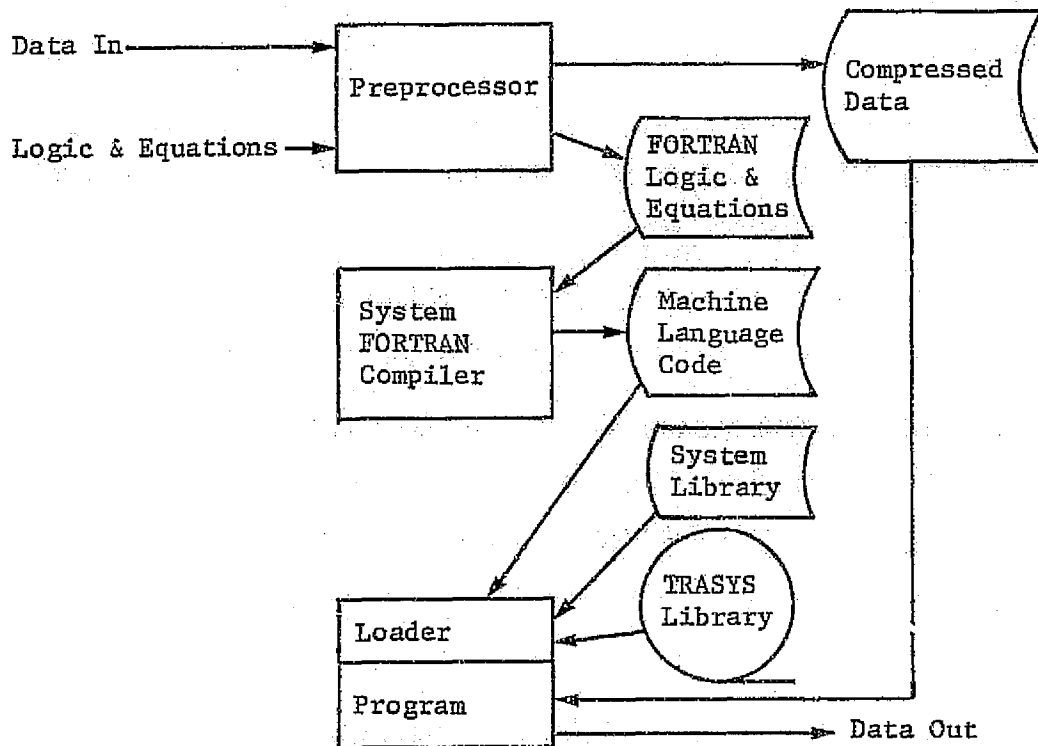


Figure I-3 Internal Flow of TRASYS

When using the full capability of TRASYS, the user will be required to exert a programming effort of sorts, in a language consisting of FORTRAN statements and problem-oriented TRASYS statements that are FORTRAN-related. This, together with the wide variety of options and features offered by the system, suggests an appropriate word of caution: TRASYS is a comprehensive system that cannot be mastered overnight. Nevertheless, to help the novice user, we have attempted to default much of the required input to normally used values so that the user need not define them.

## II. PART 1 - PREPROCESSOR

---

### A. SEGMENT DEFINITIONS

SEGMENT NAME: TRASYS

PURPOSE: Driver segment of the TRASYS preprocessor (see Fig. II-1).

CALLING SEGMENT: None

SEGMENTS CALLED: START            LOGICO            RAPUP  
                         DATARD            TPGEN

SEGMENT NAME: START

PURPOSE: This segment is the main driver segment for the preprocessor initialization, model collector, source editor, and edit output tape generator segments.

CALLING SEGMENT: TRASYS

SEGMENTS CALLED: INITIAL            SEDIT

SEGMENT NAME: INITIAL

PURPOSE: This segment initializes the preprocessor-labeled common, writes the TRASYS banner on the output file, and reads in and processes the OPTION DATA block.

CALLING SEGMENT: START

SEGMENTS CALLED: None



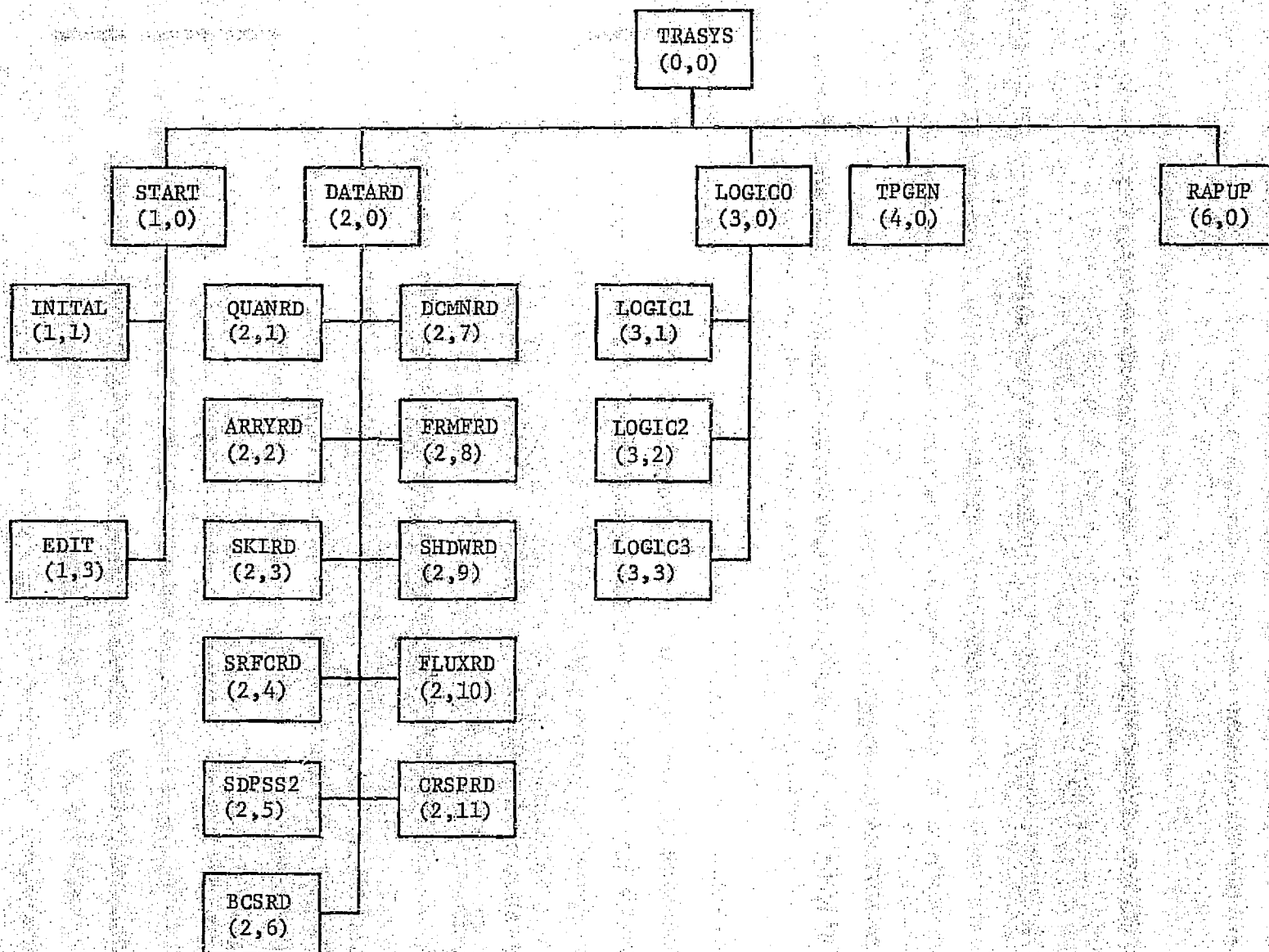


Figure II-1 Diagram of Preprocessor Segment Structure

SEGMENT NAME: SEDT

PURPOSE: This segment performs the source edit function of the preprocessor. Input data are read from the INPUT, CMERG, EMERG, and RSI files and output is written to the DATAI and RSO files.

CALLING SEGMENT: START

SEGMENTS CALLED: None

SEGMENT NAME: DATARD

PURPOSE: This segment is the main driver segment that reads in and processes the user's data input block.

CALLING SEGMENT: TRASYS

SEGMENTS CALLED:	QUANRD	SDPSS2	SHDWRD
	ARRYRD	BCSRD	FLUXRD
	SKIRD	DCMNRD	CRSPRD
	SRFCRD	FFRD	

SEGMENT NAME: QUANRD

PURPOSE: This segment reads in and processes the user's QUANTITIES DATA input block.

CALLING SEGMENT: DATARD

SEGMENTS CALLED: None

SEGMENT NAME: ARRYRD

PURPOSE: This segment reads in and processes the user's ARRAY  
DATA input block.

CALLING SEGMENT: DATARD

SEGMENTS CALLED: None

SEGMENT NAME: SKIRD

PURPOSE: This segment reads in and processes the user-input "I"  
and "K" cards of the user's SURFACE DATA input block.

CALLING SEGMENT: DATARD

SEGMENTS CALLED: None

SEGMENT NAME: SRFCRD

PURPOSE: This segment reads in the "S", "R", "D", "N", and "B" cards  
of the user's SURFACE DATA input block. These data are  
combined with the output of segment SKIRD and are output  
on file NRIOS for final processing in segment SDPSS2.

CALLING SEGMENT: DATARD

SEGMENTS CALLED: None

SEGMENT NAME: SDPSS2

PURPOSE: This segment completes the processing of the user's  
SURFACE DATA input block.

CALLING SEGMENT: DATARD

SEGMENTS CALLED: None

SEGMENT NAME: BCSRD

PURPOSE: This segment reads in and processes the user's BCS DATA input block. This segment also reads the NODE/BCS directory generated by segment SDPSS2 and processes it with the user's input data to form the communication link between the BCS data and the SURFACE data.

CALLING SEGMENT: DATARD

SEGMENTS CALLED: None

SEGMENT NAME: DCMNRD

PURPOSE: This segment reads in and writes out to the system output file the user's DOCUMENTATION DATA input block.

CALLING SEGMENT: DATARD

SEGMENTS CALLED: None

SEGMENT NAME: FFRD

PURPOSE: This segment reads in and processes the user's FORM FACTOR DATA input block.

CALLING SEGMENT: DATARD

SEGMENTS CALLED: None

SEGMENT NAME: SHDWRD

PURPOSE: This segment reads in and processes the user's SHADOW FACTOR DATA input block.

CALLING SEGMENT: DATARD

SEGMENTS CALLED: None

SEGMENT NAME: FLUXRD

PURPOSE: This segment reads in and processes the user's FLUX  
DATA input block.

CALLING SEGMENT: DATARD

SEGMENTS CALLED: None

SEGMENT NAME: CRSPRD

PURPOSE: This segment reads in and processes the user's CORRE-  
SPONDENCE DATA input block.

CALLING SEGMENT: DATARD

SEGMENTS CALLED: None

SEGMENT NAME: LOGICO

PURPOSE: This segment is the driver segment for the segments  
that read in and process the user's OPERATIONS DATA  
block.

CALLING SEGMENT: TRASYS

SEGMENTS CALLED: LOGIC1  
LOGIC2  
LOGIC3

SEGMENT NAME: LOGIC1

PURPOSE: This routine reads in the user's OPERATIONS DATA block  
and sets up all the variables needed to write the  
ODPROG segment of the processor.

CALLING SEGMENT: LOGICO

SEGMENTS CALLED: None

SEGMENT NAME: LOGIC2

PURPOSE: This routine reads the output of segment LOGIC1 and writes to the processor compile file CMPL, the main processor segment, TRASYS, and the ODPROG subsegment.

CALLING SEGMENT: LOGICO

SEGMENTS CALLED: None

SEGMENT NAME: LOGIC3

PURPOSE: This segment reads in and processes the user's SUBROUTINE DATA block and writes to the processor compiler file CMPL all subsegments of the processor that require compilation.

CALLING SEGMENT: LOGICO

SEGMENTS CALLED: None

SEGMENT NAME: TPGEN

PURPOSE: This routine writes the needed driver information of the processor to the sequential data file SQNTL.

CALLING SEGMENT: TRASYS

SEGMENTS CALLED: None

SEGMENT NAME: RAPUP

PURPOSE: This segment wraps up the preprocessor execution phase.

CALLING SEGMENT: TRASYS

SEGMENTS CALLED: None

B. SUBROUTINE AND FUNCTION DESCRIPTIONS - PREPROCESSOR

ROUTINE NAME: AAAAAA

DESCRIPTION: This routine initializes the variables containing the last program modifications number and date.

CALLING SEQUENCE: CALL AAAAAA (V, D)

V - Last version modification number

D - Date of last modification

REFERENCED BY: SEGMENT ROUTINE

INITAL INITAL

ROUTINE NAME: ABNORM1 (CDC system routine)

DESCRIPTION: This routine when called causes the program to terminate abnormally with error traceback.

CALLING SEQUENCE: CALL ABNORM1 (P1, P2, P3)

P1 - The name of the calling subroutine; left-justified, Hollerith input

P2 - A decimal number, maximum of 88, which is used as an error number. Must not be 0

P3 - The error message; left-justified zero-filled. The message must be terminated with 4 octal zeros in the rightmost position of a word

REFERENCED BY: SEGMENT ROUTINE

TRASYS	ABT1
INITAL	OPTNRD



ROUTINE NAME: ABT1

DESCRIPTION: This is an abnormal exit routine for the TRASYS pre-processor.

CALLING SEQUENCE: CALL ABT1 (N, NO)

N - Name of the routine exiting from. The name is left-justified and the rest of the field blanked within the word

NO - Type of error

- = 1 User input error
- = 2 Bad source edit input tape
- = 3 TAPE/DISK/DRUM read error
- = 4 Program limitations exceeded
- = 5 Job field length too short
- = 6 Programmer error

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>	<u>SEGMENT</u>	<u>ROUTINE</u>	<u>SEGMENT</u>	<u>ROUTINE</u>
	TRASYS	HCARD	SRFCRD	SRFCS1	FLUXRD	FLUXRD
		INRDD		CPDRC3	CRSPRD	CRSPRD
		TPERR		REFCD		
		SORTD				
		SERROR		DUPSRF	LOGICO	WCMN
					LOGIC1	LOGIC1
	SEDIT	SEDIT	SDPSS2	SDPSS2		LP1
		INKDPP	BCSRD	BCSRD		ORBGEN
		INRDE		BCSP1		ORBIN
	QUANRD	QUANRD		BCSP2		ORBNPL
		QUANSD	FFRD	FFPROC		ORBPL
	ARRYRD	ARRYRD	SHDWRD	SHDWRD		ORBPLC
						BLDGEN
	SKIRD	IP1		SFRSI	RAPUP	RAPUP
		IP2				
		KP2				

ROUTINE NAME: ARRAYD

DESCRIPTION: This routine reads in and processes the user array input block.

CALLING SEQUENCE: CALL ARRAYD

REFERENCED BY: SEGMENT      ROUTINE

          DATARD      DATARD

FILES: NRIO2- Writes three records to the random I/O file  
          Record 1 - Array names/input card sequence number  
          Record 2 - Array starting positions with respect to  
                    the output array common/length of the  
                    array +1  
          Record 3 - Array data  
NSC1 - Temporary file to store the array data  
NSC2 - Temporary file to store the array lengths  
NOUT - System output file

ROUTINE NAME: ASKCRD

DESCRIPTION: This routine decodes the user input source edit control cards for the preprocessor source editor.

CALLING SEQUENCE: CALL ASKCRD (IER)

          IER - Error detection flag  
                = 0 No error found  
                = 1 One or more errors found

KEY VARIABLES: Card data passed to this routine is common/CARD/variables NCDTYP and IND. Decoded information is returned in common/CARD/variables IND, NDOT, IDOT, JDOT.

REFERENCED BY: SEGMENT      ROUTINE

          SEDTT      SEDTT

FILES: NOUT - System output file

REV 1

II-11

ROUTINE NAME: AUTOCM

DESCRIPTION: This routine sets up a correspondence data file on  
NGBIRR for the automatic combination of surfaces  
generated by a user-input polygon.

CALLING SEQUENCE: CALL AUTOCM

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	POLYGN

FILES: NGBIRR

ROUTINE NAME: BANNPP

DESCRIPTION: This routine prints the banner page for the TRASYS  
preprocessor on the output file.

CALLING SEQUENCE: CALL BANNPP

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	INITAL	INITAL

FILES: NOUT - System output file

ROUTINE NAME: BCSP1

DESCRIPTION: This routine processes the first pass on the user-input BCS input data block.

CALLING SEQUENCE: CALL BCSP1 (NIX)

NIX - Last cell used in dynamic storage, blank common variable (IX)

KEY VARIABLE: IX - Data once converted from user input are passed to the BCSP2 routine via this blank common variable

REFERENCED BY: SEGMENT      ROUTINE

BCSRD      BCSR

FILES: NOUT - System output file

ROUTINE NAME: BCSP2

DESCRIPTION: This routine processes the block coordinate system input into data values and a BCS directory. Data are input to this routine in blank common generated by routine BCSP1. Data leave the routine via blank common.

CALLING SEQUENCE: CALL BCSP1 (NIX, NBSD, NBWD, NBSV, NBEV)

NIX - Length of vector data in blank common input to this routine

NBSV - Starting word of BCS data values in blank common (always 1)

NBEV - Length of BCS data values in blank common

NBSD - Starting word of BCS directory in blank common

NBWD - Length of BCS directory in blank common

REFERENCED BY: SEGMENT      ROUTINE

BCSRD      BCSR

FILES: NOUT - System output file

ROUTINE NAME: BCSRD

DESCRIPTION: This routine and the routines that it calls read in and process the user-input BCS DATA block and tie the user-input surface data to the BCS input data. Besides the user BCS data input, this routine reads a random I/O record that was output by the surface data block's pass 2 processor.

CALLING SEQUENCE: CALL BCSRD

KEY VARIABLES: NRIS - Random I/O record number of record written  
by surface data pass 2 processor  
NLIS - Length of the random I/O record

REFERENCED BY: SEGMENT      ROUTINE  
                 DATARD      DATARD

FILES: NOUT - System output file  
      NRIO - Random I/O file

ROUTINE NAME: BLDGEN

DESCRIPTION: This routine decodes the "BUILD" cards found in the operations data block and writes the corresponding calls to processor subroutines BUILDG and ADD.

CALLING SEQUENCE: CALL BLDGEN

REFERENCED BY: SEGMENT      ROUTINE  
                 LOGIC1      LP1

FILES: NOUT - System output file  
      NSC1 - Sequential scratch file

ROUTINE NAME: BOX

DESCRIPTION: This routine sets up the Euler angles, position vector, and program-compatible surface description parameters for the top of a five- or six-sided box input by the point method.

CALLING SEQUENCE: CALL BOX (P11, P12, P13, P31, P32, P33)

P11, P12, P13 - X, Y, and Z components of a vector colinear with the surface coordinate system Y-axis

P31, P32, P33 - X, Y, and Z components of a vector colinear with the surface coordinate system X-axis

REFERENCED BY: SEGMENT      ROUTINE

SDPSS2      SDTPS2

FILES: NOUT - System output file

ROUTINE NAME: BOXGEN

DESCRIPTION: This routine generates Euler angles, position vector, and surface description parameters for the sides and bottom (BOX6) of a box based on similar parameters set up by BOX for the top of the box.

CALLING SEQUENCE: CALL BOXGEN (P11, P12, P13, P31, P32, P33, PHI, PSI, OMG, ISURF)

P11, P12, P13 - X, Y, and Z components of a vector colinear with the surface coordinate system Y-axis

P31, P32, P33 - X, Y, and Z components of a vector colinear with the surface coordinate system X-axis

PHI, PSI, OMG - Euler angles necessary to rotate the ICS, BCS, or CCS into the surface coordinate system

ISURF - Counter indicating which side of box is being generated ( $1 \leq \text{ISURF} \leq 6$ )

REFERENCED BY: SEGMENT      ROUTINE

SDPSS2      SDTPS2

ROUTINE NAME: BSRGHD

DESCRIPTION: This routine performs a table lookup for exact equal compares. The table must be a doublet array and the routine uses the binary search technique.

CALLING SEQUENCE: CALL BSRGHD (NAME, ITABLE, NPOINT, MIDPT, IAN)

NAME - Independent variable name searched for  
ITABLE - Doublet table (NAME/ANSWER)  
NPOINT - Number of words in the table (2 \* number of entries)  
MIDPT - Binary midpoint of table (calculated in this routine of MIDPT = 0)  
IAN - Dependent variable found (ANSWER)  
= 0 if no match is found

REFERENCED BY: SEGMENT ROUTINE

INITAL	OPTNRD
SRFCRD	SRFCS1
	CDPRC3
FFRD	FFREAD
SHDWRD	SFTBRD
	SFRSI
FLUXRD	DIBLDR

ROUTINE NAME: CALAC

DESCRIPTION: This is a function subroutine that calculates surface and node areas.

CALLING SEQUENCE: CALAC (ILK, ALPH, BMIN, BMAX, GMIN, GMAX)

ILK - Surface type  
1 = Rectangle  
2 = Disc  
3 = Trapezoid  
4 = Cylinder  
5 = Cone  
6 = Sphere  
7 = Paraboloid

ALPH -	} Surface description parameters
BMIN -	
BMAX -	
GMIN -	
GMAX -	

REFERENCED BY: SEGMENT ROUTINE

SDPSS2	SDTPS2
--------	--------

REV 1

II-16



ROUTINE NAME: CALANG

DESCRIPTION: Given the direction cosines of SCS X and Y-axis in an ICS, BCS, or CCS, this routine calculates a set of Euler angles to rotate the ICS, BCS, or CCS into the SCS.

CALLING SEQUENCE: CALL CALANG (P11, P12, P13, P31, P32, P33, ROTZ, ROTY, ROTX)

P11, P12, P13 - X, Y, and Z components of a vector with magnitude BMAX that is colinear with the SCS Y-axis

P31, P32, P33 - X, Y, and Z components of a vector with magnitude GMAX that is colinear with the SCS X-axis

ROTZ, ROTY, ROTX - Euler angles to rotate ICS, BCS, or CCS into the SCS

REFERENCED BY: SEGMENT      ROUTINES

SDPSS2	BOX	PARAB
	BOXGEN	TRAPZ
	CONE	RECT
	CYLNR	SPHERE
	DISC	
	IMAGES	

FILES: NOUT - System output file

ROUTINE NAME: CALB

DESCRIPTION: This routine calculates the BMIN and BMAX values for each node.

CALLING SEQUENCE: CALL CALB (DB, BETA, IB, BMAXT, NVB, IG, DVMUL)

DB - Measure of node width in the beta direction  
BETA - Measure from the edge of the surface to the center of the current node in the beta direction  
IB - Sequence number of current node in the beta direction  
BMAXT - Temporary storage of BMAX value for surface  
NVB - Number of nodes in the beta direction  
IG - Sequence number of current row of nodes in the gamma direction  
DVMUL - Length unit multiplier from a D-card

REFERENCED BY: SEGMENT      ROUTINE

SDPSS2	SDTFS2
--------	--------

FILES: NOUT - System output file

REV 1

II-17

ROUTINE NAME: CALG

DESCRIPTION: This routine calculates the GMIN and GMAX values for each node.

CALLING SEQUENCE: CALL CALG (DG,GAMMA,IG, GMAXT,NVG,DVMUL)

DG - Measure of node length in gamma direction  
GAMMA - Measure from the beginning edge of the surface to the center of the current node in the gamma direction  
IG - Sequence number of the current node in the gamma direction  
GMAXT - Temporary storage of GMAX value for surface  
NVG - Number of nodes in gamma direction  
DVMUL - Length unit multiplier from a D-card

REFERENCED BY: SEGMENT ROUTINE

SDPSS2 SDTPS2

FILES: NOUT - System output file

ROUTINE NAME: CDPRC1

DESCRIPTION: This routine performs the first-pass conversion of a card data input that was read using an "A" format to Hollerith, integer, and floating-point words

CALLING SEQUENCE: CALL CDPRC1

KEY VARIABLES: IND - Array containing the card image data in Hollerith format  
NDOT - Array containing the converted values  
IDOT - Array describing what is contained in the NDOT array.  
JDOT - Column location in which the values in the NDOT array started on with respect to the IND array  
ID - A single word containing the number of words used in the NDOT array

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>	<u>SEGMENT</u>	<u>ROUTINE</u>
	TRASYS	HCARD	SHDWRD	SHDWRD
	QUANRD	QUANRD		STTBRD
	ARRYRD	ARRYRD	FLUXRD	FLUXRD
	SKIRD	SKIRD		DEBLDR
		KP1	CRSPRD	CRSPRD
		IP1	LOGIC1	LP1
				BLDGEN
		SRFCRD	SFRCRD	ORBGEN
		SRFCS1	LOGIC2	LOGIC2
		SRFCBC	LOGIC3	LOGIC3
	BCSRD	BCSPI		
	FFRD	FFREAD		
		FFRD		

FILES: NOUT - System output file

ROUTINE NAME: CDPRC3

DESCRIPTION: This routine compares all Hollerith words in the NDOT array to a directory array residing in the dynamic storage area for a matching name. If it finds one, then the name in the NDOT array is replaced by the corresponding data value or values. This routine should be called after CDPRC1 and before CDPRC2.

CALLING SEQUENCE: CALL CDPRC3

KEY VARIABLES: NDOT - Output array from CDPRC1 and input for this routine.  
Also the output from this routine  
IDOT & JDOT - Same as NDOT  
IX - Dynamic storage (blank common), containing the constant directory and constant data  
IXSD - Index pointing to first word of directory  
IXWD - Length of directory  
IXSV - Index pointing to first word of data values

REFERENCED BY: SEGMENT ROUTINE  
SRFCRD SRFC31

FILES: NOUT - System output file

ROUTINE NAME: CHEC

DESCRIPTION: This routine checks the validity of surface description parameter values.

CALLING SEQUENCE: CALL CHEC

REFERENCED BY: SEGMENT ROUTINE  
SDPSS2 SDTPS2

FILES: NOUT - System output file

ROUTINE NAME: CLEAR

ENTRY POINT: CHECTP

ROUTINE NAME: CDPRC2

DESCRIPTION: This routine processes the arithmetic calculations (\*, /, -, +) within a data field input on a user input card. This routine is usually called after subroutine CDPRC1, but in some instances it is called after CDPRC3.

CALLING SEQUENCE: CALL CDPRC2

KEY VARIABLES: NDOT, IDOT, JDOT and ID have the same function in this routine as they do in CDPRC1. This routine may or may not condense the arrays, depending on the type of input.

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	TRASYS	HCARD
	QUANRD	QUANRD
	ARRAYD	ARRAYD
	SKIRD	SKIRD
		KP1
		IP1
	SRFCRD	SRFCRD
		SRFCS1
		SRFCBC
	BCSRD	BCSP1
	FFRD	FFRD
		FFREAD
	SHDWRD	SHDWRD
		SFTBRD
	FLUXRD	FLUXRD
		DIBLDR
	CRSPRD	CRSPRD
	LOGIC1	ORBGEN
		BLDGEN

FILES: NOUT - System output file

ROUTINE NAME: CLEAR

DESCRIPTION: This routine frees the buffers and rewinds the unit so that it may be re-used. (Generally used on scratch files.)

CALLING SEQUENCE: CALL CLEAR (IUNIT)

IUNIT - Unit number to be freed.

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	FFRD	FFRPROC
	SRFCRD	SRFCRD

ROUTINE NAME: WFFCOM

DESCRIPTION: This routine name is an entry point into the routine WCOM. When this entry point name is called, the Fortran-labeled common for the processor FFPROG segment is written to the NCML file.

CALLING SEQUENCE: CALL WFFCOM

REFERENCED BY: SEGMENT      ROUTINE

LOGICO	WPROG
LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WSFCOM

DESCRIPTION: This routine name is an entry point into the routine WCOM. When this entry point name is called, the Fortran-labeled common for the processor SFPROG segment is written to the NCML file.

CALLING SEQUENCE: CALL WSFCOM

REFERENCED BY: SEGMENT      ROUTINE

LOGICO	WPROG
LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WNPCOM

DESCRIPTION: This routine name is an entry point into the routine WCOM. When this entry point is called, the Fortran-labeled common for the processor NPPROG segment is written to the NCML file.

CALLING SEQUENCE: CALL WNPCOM

REFERENCED BY: SEGMENT      ROUTINE

LOGIC0	WPROG
LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WOPCOM

DESCRIPTION: This routine name is an entry point into the routine WCOM. When this entry point name is called, the Fortran-labeled common for the processor OPPROG segment is written to the NCML file.

CALLING SEQUENCE: CALL WOPCOM

REFERENCED BY: SEGMENT      ROUTINE

LOGIC0	WPROG
LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WDICOM

DESCRIPTION: This routine name is an entry point into the routine WCOM. When the entry point is called, the Fortran-labeled common for the processor DIPROG segment is written to the NCML file.

CALLING SEQUENCE: CALL WDICOM

REFERENCED BY: SEGMENT      ROUTINE

LOGIC0	WPROG
LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to



ROUTINE NAME: WGBCOM

DESCRIPTION: This routine name is an entry point into the routine WCOM. When this entry point name is called, the Fortran-labeled common for the processor GBPROG segment is written to the NCML file.

CALLING SEQUENCE: CALL WGBCOM

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGICO	WPROG
	LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WCMCOM

DESCRIPTION: This routine name is an entry point into the routine WCOM. When this entry point name is called, the Fortran-labeled common for the processor CMPROG segment is written to the NCML file.

CALLING SEQUENCE: CALL WCMCOM

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGICO	WPROG
	LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WAQCOM

DESCRIPTION: This routine name is an entry point into the routine WCOM. When this entry point name is called, the Fortran-labeled common for the processor AQPROG segment is written to the NCML file.

CALLING SEQUENCE: CALL WAQCOM

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGICO	WPROG
	LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WQOCOM

DESCRIPTION: This routine name is an entry point into the routine WCOM. When this entry point name is called, the Fortran-labeled common for the processor QOPROG segment is written to the NCML file.

CALLING SEQUENCE: CALL WQOCOM

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGICO	WPROG
	LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WRBCOM

DESCRIPTION: This routine name is an entry point into the routine WCOM. When the entry point name is called, the Fortran-labeled common for the processor RBPROG segment is written to NCML file.

CALLING SEQUENCE: CALL WRBCOM

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGICO	WPROG
	LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WRCCOM

DESCRIPTION: The routine name is an entry point into the routine WCOM. When this entry point name is called, the Fortran-labeled common for the processor RCPROG segment is written to the NCML file.

CALLING SEQUENCE: CALL WRCCOM

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGICO	WPROG
	LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WDRCOM

DESCRIPTION: This routine name is an entry point into the routine WCOM. When this entry point name is called, the Fortran-labeled common for the processor DRPROG segment is written to the NCMPPL file.

CALLING SEQUENCE: CALL WDRCOM

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGICO	WPROG
	LOGIC3	LOGIC3

FILES: NCMPPL - The file that the generated processor Fortran cards are written to

ROUTINE NAME: CONE

DESCRIPTION: This routine converts point input for cones to program-compatible surface description parameters and sets up Euler angles and a position vector to transform the ICS, BCS, or CCS into the SCS of the cone.

CALLING SEQUENCE: CALL CONE

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	SDTPS2

FILES: NOUT - System output file

ROUTINE NAME: CRSPRD

DESCRIPTION: This routine and the routines that it calls read in and process the user's CORRESPONDENCE DATA block.

CALLING SEQUENCE: CALL CRSPRD

REFERENCE:	<u>SEGMENT</u>	<u>ROUTINE</u>
	DATARD	DATARD

FILES: NGBIRR - Contains the correspondence data output from this routine  
for the processor phase.  
NOUT - System output file

ROUTINE NAME: CYLNRD

DESCRIPTION: This routine converts point input for cylinders to program-compatible surface description parameters and sets Euler angles and a position vector to transform the ICS, BCS, or CCS into the SCS of the cylinder.

CALLING SEQUENCE: CALL CYLNRD

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	SDTPS2

FILES: NOUT - System output file

ROUTINE NAME: DATARD

DESCRIPTION: This routine is the driving segment that calls the other segments that read in and process the data blocks of the user's input data deck. It does not read in the logic blocks.

CALLING SEQUENCE: CALL DATARD

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	TRASYS	TRASYS

FILES: NOUT - System output file

ROUTINE NAME: DATE (CDC system routine)

DESCRIPTION: This routine returns the current date that the job was run on the computer. The date is returned in display code in the following format....

bmm/DD/YY.

CALLING SEQUENCE: CALL DATE (DTE)

DTE - Returned data variable

REFERENCED BY: SEGMENT      ROUTINE

INITAL      INITIAL

ROUTINE NAME: DCMNRD

DESCRIPTION: This routine processes the user-input documentation data block.

CALLING SEQUENCE: CALL DCMNRD

REFERENCED BY: SEGMENT      ROUTINE

DATARD      DATARD

FILES: NOUT - System output file

ROUTINE NAME: DDUMP

DESCRIPTION: This routine prints consecutive core memory words to the system output file in octal format. This routine is used to dump the error trace information of the pre-processor.

CALLING SEQUENCE: CALL DDUMP (ISADD, LEADD, NN)

ISADD - Address of word to start dumping  
LEADD - Address of word to end dumping  
NN - Integer number to print to identify the printed dump

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>	<u>SEGMENT</u>	<u>ROUTINE</u>	<u>SEGMENT</u>	<u>ROUTINE</u>
	SEDIT	SEDIT	SKIRD	SKIRD	CRSPRD	CRSPRD
		EPTAPE		IP2		
		OTWTE		KP2	LOGIC1	LOGIC1
		MERGE	SRFCRD	SRFCRD		
	QUANRD	QUANRD		SRFCS1		LP1
			BCSRD	BCSRD	TPGEN	TPGEN
		QUANS				
	ARRYRD	ARRYRD	SHDWRD	SHDWRD	FFRD	FFRD
				SFTBRD		FFPROC
				SFRB		FFREAD
				SFWPL	TRASYS	TRASYS
				SFCIO		

FILES: NOUT - System output file

ROUTINE NAME: DIBLDR

DESCRIPTION: This routine process one set of user's flux data input in the FLUX DATA input block. The output of this routine is the flux data restart file, NDIR.

CALLING SEQUENCE: CALL DIBLDR (NODIN, IQDS, IQDR, IQDP, NN, NN2, INIT, ISTEP)

NODIN - Starting location for the unsorted node array  
IQDS - Starting location for the solar data  
IQDR - Starting location for the albedo data  
IQDP - Starting location for the planetary data  
NN - Number of nodes in the node array  
NN2 - Number of words in the node directory (2 \* NN)  
INIT - Value to initialize the IQDS, IQDR, IQDP arrays to  
ISTEP - Logic block step number to which this set of flux data pertains

REFERENCES:	<u>SEGMENT</u>	<u>ROUTINE</u>
	FLUXRD	FLUXRD

FILES: NOUT - System output file  
NDIR - Flux data restart file

ROUTINE NAME: DIRCS

DESCRIPTION: Given the Euler angles necessary to rotate one coordinate system into another, this routine calculates the corresponding direction cosines.

CALLING SEQUENCE: CALL DIRCS (II, JJ, KK, PHI, PSI, OMI, TRAN)

II, JJ, KK - Integers (1, 2, or 3) defining the order in which the rotations PHI, PSI, and OMI are to be performed  
PHI, PSI, OMI - Euler angles defining the rotations about the Z, Y, and X-axes, respectively  
TRAN - A three-by-three matrix of direction cosines

REFERENCES:	<u>SEGMENTS</u>	<u>ROUTINES</u>
	SKIRD	IP1
	SDPSS2	SDTPS2
		BOX
		IMAGES

ROUTINE NAME: DISC

DESCRIPTION: This routine converts point input for discs to program-compatible surface description parameters and sets up Euler angles and a position vector to transform the ICS, BCS, or CCS into the SCS of the disc.

CALLING SEQUENCE: CALL DISC

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	SDTPS2

FILES: NOUT - System output file

ROUTINE NAME: DUPSRF

DESCRIPTION: This routine duplicates previously input surfaces and modifies surface description parameters as specified by the user.

CALLING SEQUENCE: CALL DUPSRF (ISSX, IESX, IER)

ISSX - The index of the starting location where previously input surface descriptions are loaded into blank common  
IESX - The index of the location of the end of the surface description data in blank common for a previously input surface  
IER - An indicator of errors encountered

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SRFCRD	SRFCS1

FILES: NOUT - System output file  
NRIOS - Scratch random access file



ROUTINE NAME: EPTAPE

DESCRIPTION: This routine positions the NRSI file at the beginning of the requested model so that source editing of the user's data can begin.

CALLING SEQUENCE: CALL EPTAPE (NU, NAME, IER)

NRSI - Restart input file  
NAME - Name of model requested for editing  
IER - Error return flag  
= 0 Ready for editing.  
≠ 0 Not ready for editing

REFERENCED BY: SEGMENT ROUTINE

SEDTT SEDIT

FILES: NOUT - System output file  
NRSI - Restart input file

ROUTINE NAME: FDPRC

DESCRIPTION: This routine does the \*\*, \*, /, +, - calculations between constants input within a single data field on a user-input data card.

CALLING SEQUENCE: CALL FDPRC (IPL, IPR, NWP)

IPL - Starting location within NDOT array to start performing calculations  
IPR - Ending location to stop processing calculations  
NWP - Number of words left after calculation processing. that contains data within the data field processed

REFERENCED BY: SEGMENT ROUTINE

TRASYS GDPRG2

FILES: NOUT - System output file

ROUTINE NAME: FFPROC

DESCRIPTION: This routine reads in and processes cards in the form factor data block and writes the information to the NFFR file.

CALLING SEQUENCE: CALL FFPROC (NN2,ISKIP)

NN2 - Twice the number of nodes in the surface data block  
ISKIP - Flag to indicate form factor data processing was skipped due to insufficient field length.

REFERENCED BY: SEGMENT ROUTINE

FFRD FFRD

FILES: NOUT - System output file  
NRIOS - Random access data file  
NFFR - Form factor data file

ROUTINE NAME: FFRD

DESCRIPTION: This routine decodes all form factor data block input cards.

CALLING SEQUENCES: CALL FFRD

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	DATARD	DATARD

FILES: NOUT - System output file  
NFFR - Form factor data file

ROUTINE NAME: FFREAD

DESCRIPTION: This routine decodes form factor data cards that contain node-to-node form factor data.

CALLING SEQUENCE: CALL FFREAD (NN2,NODATA,IERCOR)

NN2 - Twice the number of nodes in the surface data block  
NODATA - Counter for number of new data entries written to NSC1  
IERCOR - Counter for number of card input errors found

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	FFRD	FFPROC

FILES: NOUT - System output file  
NSC1 - Sequential scratch file  
NFFR - Form factor data file

ROUTINE NAME: FFSORT

DESCRIPTION: This routine sorts the form factor input data into the order expected by the form factor calculation segment.

CALLING SEQUENCE: CALL FFSORT (IA, IB, NN)

IA - Unsorted array

IB - Sorted array

NN - Number of values in IA and IB

REFERENCED BY:     SEGMENT     ROUTINE

FFRD

FFPROC

FILES: None

ROUTINE NAME: FLUXRD

DESCRIPTION: This routine and the routines that it calls read in and process the user's input FLUX DATA block.

CALLING SEQUENCE: CALL FLUXRD

REFERENCED BY:     SEGMENT     ROUTINE

DATARD

DATARD

FILES: NOUT - System output file

ROUTINE NAME: GNRSO

DESCRIPTION: This routine generates the first file of the RSO (restart) tape.

CALLING SEQUENCE: CALL GNRSO

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SEDIT	SEDIT

FILES: NRSO - Restart output tape

ROUTINE NAME: HCARD

DESCRIPTION: This routine decodes the header cards of the user's input data deck when the card is read from the NDI file.

CALLING SEQUENCE: CALL HCARD (N)

N = 1 Card to be processed has been read prior to  
calling this routine  
N = 0 Card is to be read before processing

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	START	HEADCD
	INITAL	OPTNRD
	SEDIT	SEDIT
	DATARD	DATARD
	QUANRD	QUANRD
	ARRYRD	ARRYRD
	SKIRD	SKIRD
	SRFCRD	SRFCRD
		SRFCS1
		SRFCBC
	BCSRD	BCSRD
	DCMNRD	DCMNRD
	FFRD	FFRD
		FFREAD
	SHDWRD'	SHDWRD
		SFTBRD
	FLUXRD	FLUXRD
		DIBLDR
	CRSPRD	CRSPRD
	LOGIC1	LP1
		BLDGEN
	LOGIC3	LOGIC3

FILES: NOUT - System output file

REV 1  
II-36

ROUTINE NAME: HEADCD

DESCRIPTION: This routine decodes the HEADER cards of the user input data deck when the card is to be read directly from the system input file.

CALLING SEQUENCE: CALL HEADCD (N)

N = 1 Card to be processed has been read prior to  
calling this routine  
N = 0 Card is to be read before processing

REFERENCED BY: SEGMENT ROUTINE  
INITAL INITAL

FILES: NIN ~ System input file  
NOUT ~ System output file

ROUTINE NAME: HTI

DESCRIPTION: This routine converts a Hollerith word containing a positive integer number to an integer number.

CALLING SEQUENCE: CALL HTI (IN, IOUT, IFLAG)

IN - Word to be converted  
IOUT - Converted word  
IFLAG - = 0 Conversion was complete  
= 1 Conversion was not complete

REFERENCED BY: SEGMENT ROUTINE  
INITAL OPTNRD

ROUTINE NAME: IMAGES

DESCRIPTION: This routine images previously input surfaces in reference planes that have been set up by subroutine REFCD and writes the surface description data for the images on the random access file.

CALLING SEQUENCE: CALL IMAGES (ISAVE, NOSF, NOND, IMAG, IREF, NLEN, INDEXS, INDEXN, IER)

ISAVE - An array of previously input surface numbers and their corresponding random access file record numbers  
NOSF - Surface counter (sequence number)  
NOND - Node counter (sequence number)  
IMAG - ID number of previously input surface that is to be imaged  
IREF - ID number of reference plane in which IMAG is to be imaged  
NLEN - Length of a random access record  
INDEXS - An array of random access record numbers for surfaces  
INDEXN - An array of random access record numbers for surfaces  
IER - An indicator of errors encountered

REFERENCED BY: SEGMENT ROUTINE

SDPSS2 SDTPS2

FILES: NOUT - System output file  
NRIO - Random access file.

ROUTINE NAME: INFORM

DESCRIPTION: This routine writes out the user information contained in the INFO file.

CALLING SEQUENCE: CALL INFORM

REFERENCED BY: SEGMENT ROUTINE

INITAL INITAL

FILES: NINFO - Information file  
NOUT - System output file

ROUTINE NAME: INITAL

DESCRIPTION: This routine initializes the preprocessor-labeled common,  
and calls the routines to write the TRASYS banner on the  
system output file and read in the OPTION DATA block.

CALLING SEQUENCE: CALL INITIAL

REFERENCED BY: SEGMENT      ROUTINE

START              START



ROUTINE NAME: INRDB

DESCRIPTION: This subroutine name is an entry point into routine INRDD.

CALLING SEQUENCE: CALL INRDB

REFERENCED BY: SEGMENT      ROUTINE

BCSRD	BCSRD
	BCSP1

ROUTINE NAME: INRDD

DESCRIPTION: This routine reads in the NDI file that contains the user-input data cards.

CALLING SEQUENCE: CALL INRDD  
CALL INRDB  
CALL INRDO  
CALL INRDDC  
CALL INRDSF  
CALL INRDOD

} Entry Points

REFERENCED BY: SEGMENT      ROUTINE      SEGMENT      ROUTINE

TRASYS	HCARD	SRFCRD	SRFCRD
QUANRD	QUANRD		SRFCS1
ARRAYD	ARRAYD		SRFCBC
SKIRD	SKIRD	CBSPRD	CRSPRD
	KP1		
	IP1		

FILES: NOUT - System output file  
NDI - Current model source data file

ROUTINE NAME: INRDE

DESCRIPTION: This routine reads in the NRSI file for the TRASYS source editor.

CALLING SEQUENCE: CALL INRDE

REFERENCED BY: SEGMENT      ROUTINE

SEDT	SEDT
------	------

FILES: NRSI - Input restart data file  
NOUT - System output file

ROUTINE NAME: INRDO

DESCRIPTION: This subroutine name is an entry point into routine INRDD.

CALL SEQUENCE: CALL INRDO

REFERENCED BY: SEGMENT      ROUTINE

FFRD	FFRD
	FFREAD
FLUXRD	FLUXRD
	DIBLDR
LOGIC3	LOGIC3

ROUTINE NAME: INRDOD

DESCRIPTION: This subroutine name is an entry point into routine INRDD.

CALLING SEQUENCE: CALL INRDOD

REFERENCED BY: SEGMENT      ROUTINE

LOGIC1	LPI
	BLDGEN

ROUTINE NAME: INRDPP

DESCRIPTION: This routine reads in the NDI file for the source editor print/punch routine.

CALLING SEQUENCE: CALL INRDPP

REFERENCED BY: SEGMENT      ROUTINE

SEDIT	PTPHSF
-------	--------

FILES: NDI - Input source edit data file

ROUTINE NAME: INRDSF

DESCRIPTION: This subroutine name is an entry point into the subroutine INRDD.

CALLING SEQUENCE: CALL INRDSF

REFERENCED BY: SEGMENT      ROUTINE

SHDWRD	SHDWRD
	SFTBRD

REV 1

II-41

ROUTINE NAME: IP1

DESCRIPTION: This routine performs the first-pass processing of the intermediate coordinate system input cards that the user input in the SURFACE DATA block.

CALLING SEQUENCE: CALL IP1 (NIX)

NIX - Index of the last word used in the dynamic storage array (IX) on completion of this routine.

REFERENCED BY: SEGMENT      ROUTINE

SKIRD      SKIRD

FILES: NOUT - System output file

ROUTINE NAME: IP2

DESCRIPTION: This routine performs the second-pass processing of the intermediate coordinate system input data that the user input in the SURFACE DATA block.

CALLING SEQUENCE: CALL IP2 (NIX)

NIX - Index of the last word in dynamic storage containing the data generated and passed by IP1.

KEY VARIABLES: NISV - Index of the starting word in dynamic storage that contains the output intermediate coordinate data  
NIEV - Index of the last word in dynamic storage that contains the output intermediate coordinate data  
NISD - Index of the starting word in dynamic storage that contains the output intermediate coordinate directory  
NIWD - Length of the intermediate coordinate directory

REFERENCES: SEGMENT      ROUTINE

SKIRD      SKIRD

FILES: NOUT - System output file

ROUTINE NAME: JOBNO (CDC system routine)

DESCRIPTION: This routine returns the current job number of the job as the computer knows it. The job number is returned in display code in the following format:

UUUUUCbb

where UUUUU are user-assigned and cc are computer-assigned.

CALLING SEQUENCE: CALL JOBNO (N)

N = Returned variable for job number

REFERENCED BY: SEGMENT      ROUTINE

INITAL      INITAL

ROUTINE NAME: KP1

DESCRIPTION: This routine performs the first-pass processing of the constant input cards the user inputs in the SURFACE DATA blocks.

CALLING SEQUENCE: CALL KP1 (NKREC, NKNAM, NKVAL, IBUF)

NKREC - Number of records written to the NSC1 file  
NKVAL - Total number of constant values, including the integer count  
NKNAM - The number of constant names  
IBUF - Scratch array that may or may not be used as a communication link between routine KP1 and KP2

REFERENCED BY: SEGMENT      ROUTINE

SKIRD      SKIRD

FILES: NOUT - System output file.

NSC1 - Scratch file 1 used to communicate between routines KP1 and KP2.

ROUTINE NAME: KP1E

DESCRIPTION: This name is an entry point into routine KP1 and is called to flush the IBUF array when the last constant data card has been processed.

CALLING SEQUENCE: CALL KP1E (NKREC, IBUF)

NKREC - Number of records written to the NSC1 file  
IBUF - Scratch array that may or may not be used  
as a communication link between routines  
KP1 and KP2

REFERENCED BY: SEGMENT ROUTINE

SKIRD SKIRD

FILES: NSC1 - Scratch file 1 used to communicate between routines  
KP1 and KP2

ROUTINE NAME: KP2

DESCRIPTION: This routine performs the second-pass processing of the constant input cards the user inputs in the SURFACE DATA blocks.

CALLING SEQUENCE: CALL KP2 (NKREC, NKNAM, NKVAL, IBUF)

Argument description is same as for routine KP1.

KEY VARIABLES: NKSD - Index of the starting word in dynamic storage  
that contains the output constant data  
NKWD - Length of the constants directory  
KCSV - Index of the starting word in dynamic storage  
that contains the output constant values  
NKEV - Index of the last word in dynamic storage that  
contains the output constant values

REFERENCED BY: SEGMENT ROUTINE

SKIRD SKIRD

FILES: NOUT - System output file  
NSC1 - Scratch file 1 used to communicate between routines KP1  
and KP2

ROUTINE NAME: LFILL

DESCRIPTION: This function routine left-justifies a word that is filled with zeros to the left of the data.

CALLING SEQUENCE: I = LFILL (IN, IFILL)

IN - Word to be left-justified

IFILL - The character to fill the word with right of the data

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>	<u>SEGMENT</u>	<u>ROUTINE</u>
	TRASYS		QUANRD	QUANRD
		CDPRC1	ARRYRD	ARRYRD
		PTSVER	SKIRD	SKIRD
	INITAL	OPTNRD		IP1
			SRFCRD	SRFCS1
				CDPRC3
			BCSRD	BCSRD
		H10T6		BCSP1
			SHDWRD	SHDWRD
			LOGIC1	LOGIC1
	SEDIT	SEDIT		BLDGEN
		MERGE	LOGIC3	LOGIC3
	CRSPRD	CRSPRD		
	FFRD	FFRD		

ROUTINE NAME: LLSF (CDC system routine)

DESCRIPTION: This routine is a function that left-shifts a word with wrap-a-round on the pushoff bits. It is usually a system routine.

CALLING SEQUENCE: I = LLSF (IWORD, IBITS)

IWORD - Word to be left-shifted

IBITS - Number of bits to shift

REFERENCED BY:

<u>SEGMENT</u>	<u>ROUTINE</u>	<u>SEGMENT</u>	<u>ROUTINE</u>	<u>SEGMENT</u>	<u>ROUTINE</u>
TRASYS	HCARD	SEDIT	SEDIT	BCSRD	BCSRD
	INRDD				BCSP1
	CDPRC1				BCSP2
	LFILL		ASKCRD	DCMNRD	DCMNRD
	PTSVER		PTPHSF	FRMFRD	FRMFRD
	BSRCHD	QUANRD	QUANRD		READVF
START	HEADCD	ARRYRD	ARRYRD	SHDWRD	SHDWRD
INITAL	OPTNRD	SKIRD	SKIRD		SFTBRD
	HTI		KP1	FLUXRD	FLUXRD
	H10T6	SRFCRD	SRFCRD		DIBLDR
	OPTNCV		SRFCS1	CRSPRD	CRSPRD
MCOLL	MCOLL		CDPRC3	LOGIC1	LP1
	MCOLCV	SDPSS2	SDTPS2	LOGIC2	LOGIC2
			IMAGES	LOGIC3	LOGIC3

REV 1  
II-45

ROUTINE NAME: LNCCOR

DESCRIPTION: This routine prints the core lengths of the processor segments as calculated by the preprocessor.

CALLING SEQUENCE: CALL LNCCOR

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC3	LOGIC3

ROUTINE NAME: LOCF (CDC system routine)

DESCRIPTION: This routine is a function that returns the absolute memory word address of the argument passed to it.  
The routine is usually a Fortran-callable system routine.

CALLING SEQUENCE: I = LOCF (J)

J = The word whose address is wanted

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	TRASYS	WSCOPE
		DDUMP
		BUFCOR
	START	ALTFILE
	INITAL	INITAL
	SKIRD	IP2
		KP2
	SDPSS2	SDPSS2
	BCSRD	BCSP2
	RAPUP	RAPUP

ROUTINE NAME: LOGICO

DESCRIPTION: This routine is the controlling driver for the processing of the user's logic block input

CALLING SEQUENCE: CALL LOGICO

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	TRASYS	TRASYS

FILES: NOUT - System output file

REV 1

II-46

ROUTINE NAME: LOGIC1

DESCRIPTION: This routine and the routines that it calls read in the array, quantities, and BCS directories. It then checks for duplicate names between the directories and reserve variable list and creates the common array consisting of names that will be used to write the processor common arrays. This routine also reads in the user's OPERATION DATA block and counts the step cards and the substep cards within a step. This information will be used to create the computed GO TO statements in routine LOGIC2.

CALLING SEQUENCE: CALL LOGIC1

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGICO	LOGICO

FILES: NOUT - System output file  
NRIO2 - Preprocessor and processor random I/O communication file

ROUTINE NAME: LOGIC2

DESCRIPTION: This routine generates the main program segment of the processor and also reads in the OPERATION DATA output from routine LPI and processes it into valid Fortran routines. These routines are written to file NCMPL.

CALLING SEQUENCE: CALL LOGIC2

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGICO	LOGICO

FILES: NOUT - System output file  
NSC1 - Scratch file containing user's operation data output by routine LPI  
NCMPL - The file that the generated processor Fortran cards are written to



ROUTINE NAME: LOGIC3

DESCRIPTION: This routine and the routines that it calls read in and process the user's SUBROUTINE DATA block. This routine also writes all of the primary routines of all the segments, along with the user-input subroutines to the NCMPL file.

CALLING SEQUENCE: CALL LOGIC3

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC0	LOGIC0

FILES: NOUT - System output file  
NCMPL - The file that the generated processor Fortran cards are written to

ROUTINE NAME: LP1

DESCRIPTION: This routine reads the user's OPERATION DATA block in pass 1 for the number of steps and number of substeps within a step. The ORBGEN user-input card is also expanded in this routine to standard type of input. The step directory is written to file NSQNTL.

CALLING SEQUENCE: CALL LP1

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC1	LOGIC1

FILES: NOUT -System output file  
NSC1 -Scratch file which the operation data block cards are written to for pass 2 processing  
NSQNTL-Preprocessor and processor sequential I/O communication file

ROUTINE NAME: LRSF

DESCRIPTION: This routine is a function that right-shifts a word with sign extension. This routine is usually a system function.

CALLING SEQUENCE: I = LRSF (IWORD, IBITS)

IWORD = Word to be right-shifted  
IBITS = Number of bits to shift

REFERENCED BY: SEGMENT ROUTINE

TRASYS	
	BSRCHD
INITAL	OPTNRD
QUANRD	QUANRD
ARRYRD	ARRYRD
SKIRD	KP1
SRFCRD	SRFCS1
BCSRD	BCSRD
SHDWRD	SFUNPK
	SFTBRD
FLUXRD	DIBLDR
SEDIT	MERGCD
	MERGE
FFRD	FFPROC
	FFREAD
	FFSORT
LOGIC2	LOGIC2

ROUTINE NAME: MERGCD

DESCRIPTION: This routine processes the CMERG source editor directives. Once the CMERG card is decoded, the NCMG file is searched to find the requested file and cards, and the requested cards are then written to the NDI file.

CALLING SEQUENCE: CALL MERGCD (NCARD)

NCARD = Last new edit sequence number used

REFERENCED BY: SEGMENT      ROUTINE

SEDT      SEDT

FILES: NOUT - System output file  
NCMG - User-input card image supplementary input file

ROUTINE NAME: MERGE

DESCRIPTION: This routine processes the EMERG source editor directives. Once the EMERG card is decoded, the NEMG file is searched for the requested model and cards, and the requested cards are then written to the NDI file.

CALLING SEQUENCE: CALL MERGE (NCARD)

NCARD = Last new edit sequence number used

REFERENCED BY: SEGMENT      ROUTINE

SEDT      SEDT

FILES: NOUT - System output file  
NEMG - User-input source edit-formatted supplementary input file

ROUTINE NAME: NONDF

DESCRIPTION: This function routine, when given the internal sequence number of the node, returns the node number.

CALLING SEQUENCE: I = NONDF (NN)

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	SDTPS2 IMAGES AUTOCM STUFDT

ROUTINE NAME: OPENMS (CDC system routine)

DESCRIPTION: This routine opens a mass storage random I/O file.

CALLING SEQUENCE: CALL OPENMS (U, IX, LENGTH, T)

U	- Unit designator
IX	- First word address in central memory of the array that will contain the file index
LENGTH	- Length of the index LENGTH - (no. of records +1) for number index. LENGTH - 2 * (no. of records +1) for name index
T	- T = 0 file is referenced through a number master index T = 1 file is referenced through a name master index (TRASYS uses only a numbered index)

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	INITAL	INITAL
	SRFCRD	SRFCRD
	SHDWRD	SHDWRD

ROUTINE NAME: OPTNCV

DESCRIPTION: This routine decodes the user's input cards that are input in the OPTION DATA block.

CALLING SEQUENCE: CALL OPTNCV

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	INITAL	OPTNRD

ROUTINE NAME: OPTNRD

DESCRIPTION: This routine reads in and processes the user-input OPTION DATA block.

CALLING SEQUENCE: CALL OPTNRD (11ST)

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	INITAL	INITAL

FILES: NOUT - System output file  
NIN - System input file

ROUTINE NAME: ORBGEN

DESCRIPTION: This routine and the routines that it calls process the user's logic input ORBGEN card.

CALLING SEQUENCE: CALL ORBGEN

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC1	LP1

FILES: NOUT - System output file  
NSC1 - Scratch file that the output of logic pass 1 is written to

ROUTINE NAME: ORBIN

DESCRIPTION: This routine writes the operations data block code for automatic generation of inertial (sun or star) oriented orbits.

CALLING SEQUENCE: CALL ORBIN (IREP, TRU, DVC, NPOINT, ILORC)

IREP - Repeat flag for complete orbits  
TRU - Initial true anomaly  
DV6 - True anomaly increment  
NPOINT - Number of orbit points generated  
ILORC - Flag to eliminate AQCAL calls from operations data block code

REFERENCED BY: SEGMENT ROUTINE

LOGIC1 ORBGEN

FILES: NOUT - System output file  
NSC1 - Scratch file that the output of logic pass 1 is written to

ROUTINE NAME: ORBNPL

DESCRIPTION: This routine writes the operations data block code for automatic generation of heliocentric orbits.

CALLING SEQUENCE: CALL ORBNPL (IREP, TRU, DV6, NPOINT, ILORC)

IREP - Repeat flag for complete orbits  
TRU - Initial true anomaly  
DV6 - True anomaly increment  
NPOINT - Number of orbit points generated  
ILORC - Flag to eliminate AQCAL calls from operations data block code

REFERENCED BY: SEGMENT ROUTINE

LOGIC1 ORBGEN

FILES: NOUT - System output file  
NSC1 - Scratch file that the output of logic pass 1 is written to

ROUTINE NAME: ORBPL

DESCRIPTION: This routine writes the operations data block code for automatic generation of noncircular, planet-oriented orbits.

CALLING SEQUENCE: CALL ORBPL (IREP, TRU, DV6, NPOINT, ILORC)

IREP	}	Reference ORBIN
TRU		
DV6		
NPOINT		
ILORC		

REFERENCED BY: SEGMENT      ROUTINE

LOGIC1      ORBGEN

FILES: NOUT - System output file  
NSC1 - Scratch file that the output of logic pass 1 is written to

ROUTINE NAME: ORBPLC

DESCRIPTION: This routine writes the operations data block code for automatic generation of circular, planet-oriented orbits.

CALLING SEQUENCE: CALL ORPLC (IREP, TRU, DV6, NPOINT, ILORC)

IREP	}	Reference ORBIN
TRU		
DV6		
NPOINT		
ILORC		

REFERENCED BY: SEGMENT      ROUTINE

LOGIC1      ORBGEN

FILES: NOUT - System output file  
NSC1 - Scratch file that the output of logic pass 1 is written to

ROUTINE NAME: OTWTE

DESCRIPTION: This routine writes the NDI file called by the TRASYS source editor.

CALLING SEQUENCE: CALL OTWTE (IFLUSH)

IFLUSH = 0 Not the last call to this routine  
IFLUSH = 1 The last call to this routine. Flush the IOB array to the NDI file

KEY VARIABLES: IOB - An array used for collecting card images until IOT words have been filled  
IOT - Last word to fill in the IOB array before flushing the data to the NDI file

REFERENCED BY: SEGMENT      ROUTINE  
  
                 SEDIT            SEDIT  
                                  MERGE  
                                  MERGCD

FILES: NDI - Source edit-formatted output file used as input file for data and logic preprocessor segments

ROUTINE NAME: PAGEH

DESCRIPTION: This routine ejects the page and writes the page heading on all pages generated by the preprocessor.

CALLING SEQUENCE: CALL PAGEH (N)

N = 0 Do not write card column designator line  
N = 1 Write card column designator line

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>	<u>SEGMENT</u>	<u>ROUTINE</u>
	TRASYS	INRDD	SRFCRD	SRFCS1
		SERROR		SRFCBC
	START	HEADCD	BCSRD	BCSRD
	INITAL	OPTNRD	DCMNRD	DCMNRD
		INFORM		
			FFRD	FFRD
	SEDT	SEDT	SHDWRD	SHDWRD
			FLUXRD	FLUXRD
		MERGE	CRSPRD	CRSPRE
		PTPHSF	LOGIC1	LOGIC1
		MERGCD	LOGIC2	LOGIC2
		PTHSTY	LOGIC3	LOGIC3
			LNGCOR	
	DATARD	DATARD	RAPUP	RAPUP
	QUANRD	QUANRD		
	ARRYRD	ARRYRD		
	SKIRD	SKIRD		
		PTCKS1		

FILES: NOUT - System output file

REV 1  
II-55



ROUTINE NAME: PARAB

DESCRIPTION: This routine converts point input for paraboloids to program-compatible surface description parameters and sets up Euler angles and a position vector to transform the ICS, BCS, or CCS into the SCS of the paraboloid.

CALLING SEQUENCE: CALL PARAB

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	SDTPS2

FILES: NOUT - System output file

ROUTINE NAME: PDUMP (CDC system routine)

DESCRIPTION: This routine dumps the main memory on the system output file.

CALLING SEQUENCE: CALL PDUMP (A, B, C)

A - Starting location to start dumping  
B - Last word to be dumped  
C - Mode in which to dump the words  
    0 = Octal dump  
    1 = Real dump  
    2 = Integer dump  
    3 = Octal dump  
    4 = Octal dump

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	SDTPS2
	FRMFRD	IMAGES FRMFRD BLDR FINAL MERGF
	FLUXRD	FLUXRD DIBLDR

FILES: NOUT - System output file

ROUTINE NAME: POLYGN

DESCRIPTION: This routine generates  $N - 2$  triangles from point input, where  $N$  is the number of points, and sets up Euler angles and a position vector for each triangle generated.

CALLING SEQUENCE: CALL POLYGN (ISURF)

ISURF - A counter, from 1 to  $N-2$ , indicating which triangle is being generated

REFERENCED BY: SEGMENT      ROUTINE

SDPSS2      SDTPS2

FILES: NOUT - System output file

ROUTINE NAME: POSIT

DESCRIPTION: This routine transposes a vector in the ICS, BCS, or CCS into the SCS of a reference plane (for imaging purposes), negates the Z component, and transforms it back into the ICS, BCS, or CCS.

CALLING SEQUENCE: CALL POSIT (X, Y, Z, TRAN)

X, Y, Z - Vector components in the ICS, BCS, or CCS  
TRAN - A 3x3 matrix of direction cosines to transform a vector from the ICS, BCS, or CCS to the SCS

REFERENCED BY: SEGMENT      ROUTINE

SDPSS2      IMAGES

ROUTINE NAME: PPTIM (CDC system routine)

DESCRIPTION: This routine will return the accumulated peripheral processor time, in integer seconds, incurred up to the time of the call.

CALLING SEQUENCE: CALL PPTIM (N)

N = Return integer preprocessor time

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	SLTPS2
	SDPSS2	IMAGES
	TRASYS	TRASYS
	START	START
	DATARD	DATARD
	LOGICO	LOGICO
	TPGEN	TPGEN
	RAPUP	RAPUP

ROUTINE NAME: PRNTCK

DESCRIPTION: This routine outputs the traced node/surface data if the variable ITRC25 is set in the OPTIONS DATA block.

CALLING SEQUENCE: CALL PRNTCK (ITP)

ITP - Flag defining node dump or surface dump

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	SDTPS2
		IMAGES

FILES: NOUT - System output file

ROUTINE NAME: PTCHEK

DESCRIPTION: This routine, given an array of points, checks for duplicates, and returns ICHEK as a flag.

CALLING SEQUENCE: CALL PTCHK (N, IT, JT, X, ICHEK)

N - Number of points to be checked  
IT - Skip this point  
JT - Skip this point  
X - Array of point data  
ICHEK - 0 = Points not same  
1 = Points same

REFERENCED BY: SEGMENT ROUTINE

SDPSS2 BOX  
POLYGN  
TRAPZ  
RECT

ROUTINE NAME: PTCKS1

DESCRIPTION: This routine writes out the surface ICS directory, ICS values, constant-directory and the constant values.  
This routine is only executed when the surface data pass  
1 error trace flag is turned on in the OPTIONS DATA block.

CALLING SEQUENCE: CALL PTCKS1

REFERENCED BY: SEGMENT ROUTINE

SKIRD SKIRD

FILES: NOUT - System output file

ROUTINE NAME: PTDIR

DESCRIPTION: This routine writes the directory record of a source edit-formatted tape to the system output file.

CALLING SEQUENCE: CALL PTDIR (NU, IDIR)

NU - File that contained the directory to be printed

IDIR - The array containing the directory to be printed

FILES: NOUT - System output file

ROUTINE NAME: PTHSTY

DESCRIPTION: This routine writes the history record of a source edit-formatted tape to the system output file.

CALLING SEQUENCE: CALL PTHSTY (IHS, ITTPFT)

IHS - The array that contains the history record to be printed

ITTPFT - Approximate tape footage used by model being printed

REFERENCED BY: SEGMENT ROUTINE

GNEDO GNEDO

FILES: NOUT - System output file

ROUTINE NAME: PTPHSF

DESCRIPTION: This routine prints and/or punches the NDI source edit-formatted file.

CALLING SEQUENCES: CALL PTPHSF (ISEQ)

ISEQ = 0      No sequencing in Columns 73-80 is wanted  
                 on the punched output  
ISEQ  $\neq$  0      Sequencing in Column 73-80 is wanted on  
                 the punched output

REFERENCED BY: SEGMENT      ROUTINE  
                 SEDIT          SEDIT

FILES: NOUT - System output file

ROUTINE NAME: PTSVER

DESCRIPTION: This routine writes the character mark under the bad character or field when an error is encountered. This routine also saves the bad card edit sequence line number so that it can be printed at the end of the preprocessor execution.

CALLING SEQUENCE: PTSVER (IC, IER)

IC - Column to print error character mark  
     IC = 1 is actual character 7 on the input card.  
IER - = 0 Fatal error was encountered.  
      $\neq$  0 Caution error was encountered.

REFERENCED BY: SEGMENT      ROUTINE  
                 TRASYS          SERROR

FILES: NOUT - System output file

ROUTINE NAME: QUANRD

DESCRIPTION: This routine processes the user-input QUANTITIES DATA block.

CALLING SEQUENCE: CALL QUANRD

REFERENCED BY: SEGMENT     ROUTINE  
                  DATARD        DATARD

FILES: NOUT - System output file  
      NSC1 - Scratch file used to write the user quantity values.  
            This file is written in pass 1 and read in pass 2  
      NR10 - Preprocessor and processor random I/O communication  
            file

ROUTINE NAME: QUANSD

DESCRIPTION: This routine writes to the NR10 file the control quantities directory and values when the user does not input a QUANTITIES DATA block.

CALLING SEQUENCE: CALL QUANSD

REFERENCED BY: SEGMENT     ROUTINE  
                  QUANRD        QUANRD

FILES: NOUT - System output file  
      NR10 - Preprocessor and processor random I/O communication file

ROUTINE NAME: RAPUP

DESCRIPTION: This routine writes the time and core statistics at the end of the preprocessor output. This routine also terminates the preprocessor either normally or in the case of fatal errors, abnormally.

CALLING SEQUENCE: CALL RAPUP

REFERENCED BY: SEGMENT     ROUTINE  
                  TRASYS        TRASYS

FILES: NOUT - System output file

ROUTINE NAME: RDPSN

DESCRIPTION: This routine, given the dimension of a node, defines the radius of a sphere large enough to encompass the node and the position vector to the center of the sphere.

CALLING SEQUENCE: CALL RDPSN (VEC, BETA, GAMMA, DB, DG)

VEC(3) - Array containing (X, Y, Z) position to center  
BETA, GAMMA - Length along side of node  
DB, DG - Distance to center of node

REFERENCED BY: SEGMENT      ROUTINE

SDPSS2      SDTPS2

ROUTINE NAME: RD PSS

DESCRIPTION: This routine computes the radius and corresponding position vector used to encompass the surface and locate the center.

CALLING SEQUENCE: CALL RD PSS (VEC)

VEC - Array of position vectors

REFERENCED BY: SEGMENT      ROUTINE

SDPSS2      SDTPS2

ROUTINE NAME: READMS

DESCRIPTION: This routine reads a logical record from a random I/O mass storage file.

CALLING SEQUENCE: CALL READMS (U, FWA, N, K)

U - Unit designator.  
FWA - Address in central memory of first word of record  
N - Number of central memory words in the record to be transferred  
K - Number index of record or name index of record to be read. (In TRASYS, K = a number index always)

REFERENCE BY: SEGMENT      ROUTINE

SRFCRD      DUPSRF  
SDPSS2      IMAGES  
BCSRD      BCSR  
SHDWRD      SFWPL  
LOGIC1      LOGIC1



ROUTINE NAME: READVF

DESCRIPTION: This routine reads in header form factor data and loads the various arrays generating all requests for data.

CALLING SEQUENCE: CALL READVF (NODIN1, JEOF, INDEX, NN2, NN, IFRST, NMAX)

NODIN1 - Array of packed node numbers and values  
JEOF - = 3HEND End of data, ≠ 3HEND Not end of data  
INDEX - Number of values stored in NODIN1  
NN2 - Number of nodes \* 2  
NN - Number of nodes  
IFRST - Flag indicating first time called  
NMAX - Maximum number of words in data block

REFERENCED BY: SEGMENT ROUTINE

FRMFRD BLDR

FILES: NOUT - System output file

ROUTINE NAME: RECT

DESCRIPTION: This routine converts point input for rectangles to program-compatible surface description parameters and sets up Euler angles and a position vector to transform the ICS, BCS, or CCS into the SCS of the rectangle.

CALLING SEQUENCE: CALL RECT

REFERENCED BY: SEGMENT ROUTINE

SDPSS2 SDTPS2

FILES: NOUT - System output file

ROUTINE NAME: REFCD

DESCRIPTION: This routine sets up surface description information for reference planes and writes it on the random access file.

CALLING SEQUENCE: CALL REFCD (ISSX)

ISSX - The index of the starting location where surface description data for the reference plane is to be loaded in blank common

REFERENCED BY: SEGMENT ROUTINE

SRFCRD SRFCS1

FILES: NRIO - Random access file

ROUTINE NAME: RETURNS (CDC system function)

DESCRIPTION: This routine releases a local file from a job.

CALLING SEQUENCE: CALL RETURNS (N)

N = Name of file to be released

REFERENCED BY: SEGMENT      ROUTINE

SRFCRD      SRFCRD

ROUTINE NAME: SCHECK

DESCRIPTION: This routine checks the validity of user-input surface description parameters and sets the default values.

CALLING SEQUENCE: CALL SCHECK (ISSX, IERR)

ISSX - The index of the starting location of surface description data in blank common

IERR - Counter for the number of fatal errors found in the user-input surface descriptions

REFERENCED BY: SEGMENT      ROUTINE

SRFCRD      SRFCSD

FILES: NOUT - System output file

ROUTINE NAME: SDPSS2

DESCRIPTION: This routine sets up blank common indexes for use in processing the surface description data in pass 2.

CALLING SEQUENCE: CALL SDPSS2

REFERENCED BY: SEGMENT      ROUTINE

DATARD      DATARD

FILES: NOUT - System output file

ROUTINE NAME: SDTPS2

DESCRIPTION: This routine reads user-input surface description data from a sequential scratch file that was written in surface data pass 1, converts the data to program compatible surface description parameters, and writes the data on the random access file.

CALLING SEQUENCE: CALL SDTPS2 (INDEXS, INDEXN, ISAVE)

INDEXS - Array of BCS names and corresponding random access record numbers for surfaces  
INDEXN - Array of random access record numbers for nodes  
ISAVE - Array of surface numbers and corresponding random access record numbers

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	SDPSS2

FILES: NOUT - System output file  
NRIO - Random access file  
NSCR1 - Scratch file

ROUTINE NAME: SECOND

DESCRIPTION: This routine returns the central-processor time from the start of job, in seconds, in floating-point format, accurate to one thousandth of a second

CALLING SEQUENCE: CALL SECOND (T)

T = Variable that central processor seconds will be returned to

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	TRASYS	TRASYS
	START	START
	DATARD	DATARD
	LOGICO	LOGICO
	TPGEN	TPGEN
	RAPUP	RAPUP

ROUTINE NAME: SEDIT

DESCRIPTION: This routine and the routines that it calls perform the preprocessor source edit function of the TRASYS program.

CALLING SEQUENCE: CALL SEDIT

REFERENCED BY: SEGMENT      ROUTINE

START              START

FILES: NIN - System input file  
NOUT - System output file

ROUTINE NAME: SERROR

DESCRIPTION: This routine is called when a caution or error condition results when preprocessing the user's input. This routine and the routines that it calls store the error accounting information and account for proper line-page format on printed error messages.

CALLING SEQUENCE: CALL SERROR (NTYP, NARRW, ILINE, NCOL, KER)

NTYP = 0 Caution message is to be printed  
      = 1 Error message is to be printed  
NARRW - Column to print error character  
ILINE - Number of error message lines to be printed  
NCOL = 0 No output column number on page heading  
      = 1 Print output column numbers on page heading  
KER = The returned caution or error number

REFERENCED BY: SEGMENT      ROUTINE      SEGMENT      ROUTINE

TRASYS	HCARD	SDPSS2	BOX
	CDPRC1		CALB
	CDPRC2		CALG
	FDPRC		CHEC
START	HEADCD		CONE
INITAL	OPTNRD		CYLNRD
SEDIT	SEDIT		DISC
			IMAGES
			PARAB
	EPTAPE		POLYGN
	ASKCRD		TRAPZ
	INRDE		RECT
	MERGC		SPHERE
DATARD	DATARD	BCSRD	BCSRD
	STORDT		
QUANRD	QUANRD		BCSP1
ARRYRD	ARRYRD		
			BCSP2
		FRMFRD	FRMFRD

REFERENCED BY: (continued)

<u>SEGMENT</u>	<u>ROUTINE</u>	<u>SEGMENT</u>	<u>ROUTINE</u>
SKIRD	SKIRD		FINAL READVF
	KP1		
	IP1	SHDWRD	SHDWRD
	IP2		SFTBRD
	KP2		
SRFCRD	SRFCRD	FLUXRD	FLUXRD
	SRFCS1		DIBLDR
	SRFCBC	CRSPRD	CRSPRD
	CDPRC3		
	DUPSRF	LOGIC0	LOGIC0
	SCHECK	LOGIC1	LOGIC1
			LP1
	SRFCDM		ORBGEN
		LOGIC2	LOGIC2
		LOGIC3	LOGIC3

FILES: NOUT - System output file.

ROUTINE NAME: SFCIO

DESCRIPTION: This routine copies the unused or requested SAVE models on the NSHADI file to the NSHADO file.

CALLING SEQUENCE: CALL SFCIO (ISHAD2, NSF)

ISHAD2 = Shadow factor request directory

NSF = Number of shadow factor models on NSHADI file.

REFERENCED BY: SEGMENT ROUTINE

SHDWRD SHDWRD

FILES: NSHADI - Shadow factor use-input file  
 NSHADO - Shadow factor user-output file  
 NOUT - System output file

ROUTINE NAME: SFRSI

DESCRIPTION: This routine reads in the requested shadow factor data from the NSHADI file and writes the data to the NPLSR file.

CALLING SEQUENCE: CALL SFRSI (NDE1, NDE2, NDE3, NDE4, IECC, IPCC, NODEA, NODET)

NDE1 - Last word of the IX array that contains the node array directory input by the user  
NDE2 - Last word of the IX array that contains the node array directory input from the NSHADI file  
NDE3 - Last word of the IX array that contains the shadow factor request directory  
NDE4 - Last word of the IX array that contains the scratch random I/O file index array (used on CDC systems only)  
IECC - Output array from this routine that will contain the unpacked cone-clock values  
IPCC - Array containing the cone-clock values in packed form read from the NSHADI file in this routine  
NODEA - Number of nodes in the node array input from user input data  
NODET - Number of nodes in the node array input from the NSHADI file

KEY VARIABLES: IX - Preprocessor blank common dynamic storage array

REFERENCED BY: SEGMENT      ROUTINE

SHDWRD      SHDWRD

FILES: NOUT - System output file  
NSHADI - Shadow factor user input file

ROUTINE NAME: SFTBRD

DESCRIPTION: This routine reads and processes the TABLE and RECOMP shadow factor data block input cards for a single node and writes the processed data to the NRIOS file.

CALLING SEQUENCE: CALL SFTBRD (NDE1, NDS3, IECC)

NDE1 - Last word of the IX array that contains the  
node array directory input by the user  
NDS3 - Starting address of the shadow factor request  
directory  
IECC - Scratch array to contain the cone-clock value  
for writeout to the NRIOS file

REFERENCED BY: SEGMENT ROUTINE

SHDWRD SHDWRD

FILES: NOUT - System output file  
NRIOS - Scratch random I/O file that the user-input shadow  
factor data are written to

ROUTINE NAME: SFWPL

DESCRIPTION: This routine writes the NPLSR file from the shadow factor data that reside on the NRIOS file.

CALLING SEQUENCE: CALL SFWPL (NDE1, NDS3, IECC, KECC)

NDE1 - Last word of the IX array that contains the  
node array directory input by the user  
This array is also the node output directory  
NDS3 - Starting word of the IX array that contains  
the shadow factor request directory  
IECC - Scratch array to read in and write out  
of  
KECC - Length of the records to be read off the  
NRIOS file

REFERENCED BY: SEGMENT      ROUTINE  
                 SHDWRD      SHDWRD

FILES: NOUT - System output file  
NRIOS - Scratch random I/O file  
NPLSR - File used to transmit the shadow factor data from the pre-processor to the processor

ROUTINE NAME: SHDWRD

DESCRIPTION: This routine and the routines that it calls read in and process the SHADOW DATA user's input block, combine it with the shadow factor data residing on a shadow factor input tape, and output the data on a file to be processed in the processor phase.

CALLING SEQUENCE: CALL SHDWRD

REFERENCED BY: SEGMENT      ROUTINE  
                 DATARD      DATARD

FILES: NOUT - System output file  
NPLSR - File used to transmit the shadow factor data from the preprocessor to the processor  
NSHADI- The user-input shadow factor input tape  
NRIOS - Scratch random I/O file



ROUTINE NAME: SKFILE

DESCRIPTION: This routine skips to and positions after an end-of-file mark on a specified mass storage unit.

CALLING SEQUENCE: CALL SKFILE (U, N)

U - Unit name or number that is to have the file skipped

N - The number of files to skip

REFERENCED BY: SEGMENT      ROUTINE

SEDTT	EPTAPE
GNEDO	GNEDO
DATARD	STORDT
SHDWRD	SHDWRD
	SFCIO

ROUTINE NAME: SKIRD

DESCRIPTION: This routine, along with the routines that it calls, reads in and processes the I and K cards that the user inputs in the SURFACE DATA input block.

CALLING SEQUENCE: CALL SKIRD

REFERENCED BY: SEGMENT      ROUTINE

DATARD	DATARD
--------	--------

FILES: NOUT - System output file

ROUTINE NAME: SORTD

DESCRIPTION: This routine numerically sorts a doublet array. The original order is not preserved on equal comparisons.

CALLING SEQUENCE: CALL SORTD (IA, NA)

IA - The array to be sorted. The sorted array is also returned in this array

NA - The length of the array to be sorted

REFERENCED BY: SEGMENT      ROUTINE

INITAL	OPTNRD
QUANRD	QUANRD
ARRYRD	ARRYRD
SKIRD	SKIRD
SRFCRD	SRFCSI

REFERENCED BY: (continued)

<u>SEGMENT</u>	<u>ROUTINE</u>
BCSRD	BCSRD
FRMFRD	FRMFRD
SHDWRD	SHDWRD
	SFRSI
FLUXRD	FLUXRD
LOGIC1	LOGIC1
	LPI

ROUTINE NAME: SORTDB

DESCRIPTION: This routine numerically sorts a doublet array. The original order is preserved on equal comparisons.

CALLING SEQUENCE: CALL SORTDB (IA, IB, NN)

IA - The array to be sorted. The sorted array is also returned in the array  
IB - A scratch array equal in length to the IA array  
NN - The length of the array to be sorted

<u>REFERENCED BY:</u>	<u>SEGMENT</u>	<u>ROUTINE</u>
	FRMFRD	BLDR

ROUTINE NAME: SORTS

DESCRIPTION: This routine numerically sorts an array. The original order is not preserved on equal comparisons.

CALLING SEQUENCE: CALL SORTS (A, JJ)

A - The array to be sorted. The sorted array is also returned in this array  
JJ - Length of the array to be sorted

<u>REFERENCED BY:</u>	<u>SEGMENT</u>	<u>ROUTINE</u>
	BCSRD	BCSRD

ROUTINE NAME: SPHERE

DESCRIPTION: Converts point input for spheres to program-compatible surface description parameters and sets up Euler angles and a position vector to transform the ICS, BCS, or CCS into the SCS of the sphere.

CALLING SEQUENCE: CALL SPHERE

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	SDTPS2

FILES: NOUT - System output file

ROUTINE NAME: SRFCBC

DESCRIPTION: This routine processes the BCS user-input cards input in the BCS DATA block.

CALLING SEQUENCE: CALL SRFCBC

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SRFCRD	SRFCRD

FILES: NOUT - System output file

ROUTINE NAME: SRFCDM

DESCRIPTION: This routine processes the DIMENSION variable and its associated data values that the user inputs as part of the surface description input.

CALLING SEQUENCE: CALL SRFCDM (KTYPE, IDM, JDM)

KTYPE - The type of surface being processed:

1 = Rectangle	6 = Sphere
2 = Disk	7 = Paraboloid
3 = Trapezoid	8 = 5-Sided box
4 = Cylinder	9 = 6-Sided box
5 = Cone	10 = Polygon

IDM - Length of the JDM array

JDM - Array containing the data to be processed

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SRFCRD	SRFCS1

FILES: NOUT - System output file

ROUTINE NAME: SRFCRD

DESCRIPTION: This routine and the routines that it calls read in and process the surface description data the user inputs in the SURFACE DATA block.

CALLING SEQUENCE: CALL SRFCRD

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	DATARD	DATARD

FILES: NOUT - System output file  
NRIOS - Scratch random I/O file

ROUTINE NAME: SRFCS1

DESCRIPTION: This routine reads in the S, R, N, and D cards of the user-input SURFACE DATA block, decodes these cards, and writes the surface information to file NRIOS for input to the surface data pass 2 processing.

CALLING SEQUENCE: CALL SRFCS1

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SRFCRD	SRFCRD

FILES: NOUT - System output file  
NRIOS - Temporary random I/O scratch file for passing the surface description information to the surface data pass 2 processor

ROUTINE NAME: START

DESCRIPTION: This routine calls routines that initialize the label commons, write the TRASYS preprocessor banner page on the system output file, read in and process the user-input OPTION DATA block, and perform the model collecting and/or source editing functions.

CALLING SEQUENCE: CALL START

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	TRASYS	TRASYS

ROUTINE NAME: STORDT

DESCRIPTION: This routine writes all header records to internal data files and keeps count of the internal files.

CALLING SEQUENCE: CALL STORDT (NUNIT, ISTEP, LABEL1, LABEL2)

NUNIT - Unit to contain data  
ISTEP - Step number  
LABEL1 - Identifies 1  
LABEL2 - Identifies 2

REFERENCED BY: SEGMENT      ROUTINE

SDPSS2	AUTOCD
FRMFRD	FINAL
SHDWRD	SHDWRD
FLUXRD	DIBLDR
CRSPRD	CRSPRD

ROUTINE NAME: STUFDT

DESCRIPTION: This routine stores all data directly applicable to each node/surface that requires no conversion determined in surface data pass 2.

CALLING SEQUENCE: CALL STUFDT

REFERENCED BY: SEGMENT      ROUTINE

SDPSS2	SDTPS2
--------	--------

ROUTINE NAME: TIME (CDC system routine)

DESCRIPTION: This routine will return the current clock time in Hollerith code of the format...

HH.MM.SS (CDC)	HHMMSS (UNIVAC)
----------------	-----------------

CALLING SEQUENCE: CALL TIME (TME)

TME - The returned time variable

REFERENCED BY: SEGMENT      ROUTINE

TRASYS	PAGEH
--------	-------

ROUTINE NAME: TPERR

DESCRIPTION: This subroutine writes out a message whenever a tape/  
file read error is encountered and aborts run.

CALLING SEQUENCE: CALL TPERR (ISUB,NUNIT)

ISUB - Name of subroutine where read error was  
encountered

NUNIT - Unit being read when read error was  
encountered

FILE: NOUT - System output file

ROUTINE NAME: TPGEN

DESCRIPTION: This routine writes the variable record pointers for the random I/O file and the TRASYS processor logical file designators to the NSQNTL file.

CALLING SEQUENCE: CALL TPGEN

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	TRASYS	TRASYS

FILES: NSQNTL - The preprocessor and processor communication file

ROUTINE NAME: TRAPZ

DESCRIPTION: This routine converts point input for trapezoids to program-compatible surface description parameters and sets up Euler angles and a position vector to transform the ICS, BCS, or CCS into the SCS of the trapezoid.

CALLING SEQUENCE: CALL TRAPZ (ISCS)

ISCS - Flag to indicate if points were input in the surface coordinate system

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	SDPSS2	SDTPS2

FILES: NOUT - System output file

ROUTINE NAME: TRASYS

DESCRIPTION: This routine is the major segment driver for the TRASYS preprocessor. It calls five subsegments; and initialization segment, data read segment, logic read setment, communication file initialization segment, and the wrapup segment.

CALLING SEQUENCE: None

REFERENCED BY: None (this is the main preprocessor segment.)

ROUTINE NAME: TRNGLE

DESCRIPTION: This is an entry point in subroutine TRAPZ that converts point input for triangles to program-compatible surface description parameters and sets up Euler angles and a position vector to transform the ICS, BCS, or CCS into the SCS of the triangle.

CALLING SEQUENCE: CALL TRNGLE (ISCS)

ISCS - Flag to indicate if points were input in the surface coordinate system

REFERENCED BY: SEGMENT      ROUTINES

SDPSS2      IMAGES  
             POLYGN

FILES: NOUT - System output file

ROUTINE NAME: TRS3

DESCRIPTION: This routine transforms a vector in an SCS to a vector in the ICS, BCS, or CCS.

CALLING SEQUENCE: CALL TRS3 (X, Y, Z, A, B, C, RX, RY, RZ, TRAN)

KEY VARIABLES: X, Y, Z      - Components of vector in ICS, BCS, or CCS  
                 A, B, C      - Components of vector in SCS  
                 RX, RY, RZ - Position vector of SCS origin in ICS, BCS,  
                                 or CCS  
                 TRAN      - 3x3 transformation matrix

REFERENCED BY: SEGMENT      ROUTINES

SDPSS2      BOX  
             IMAGES  
             SDTPS2



ROUTINE NAME: WCMMN

DESCRIPTION: This routine writes to file NCMPL a Fortran-labeled common statement based on the information in the argument list.

CALLING SEQUENCE: CALL WCMMN (NAME, LNM, N, NT)

NAME - Common name  
LNM - Array containing a list of names to be the variables within the common being written  
N - Length of the LNM array  
NT - Type of common to write  
= 0 Single-value variable name  
= 1 Single-dimension (vector) names. The LNM under this option is in format...  
Word 1 NAME  
2 Dimension  
3 NAME  
(etc)

REFERENCED BY: SEGMENT ROUTINE

LOGICO WCOMO

FILES: NCMPL - The file that the generated Fortran cards are written to

ROUTINE NAME: WCOMO

DESCRIPTION: This routine writes the TRASYS main-program segment commons to the NCMPL file.

CALLING SEQUENCE: CALL WCOMO

REFERENCED BY: SEGMENT ROUTINE

LOGICO WPRGO  
WRITNP  
LOGIC2 LOGIC2  
LOGIC3 LOGIC3

FILES: NCMPL - The file that the generated Fortran cards are written to

ROUTINE NAME: WCOM

DESCRIPTION: This routine is a collection of entry points that write the Fortran common statements of the processor subsegments.

CALLING SEQUENCE: WCOM is not called, only the entry points are called.

ENTRY POINTS: CALL CM30 - FFPROG segment  
CALL CM40 - SFPROG segment  
CALL CM50 - NPPROG segment  
CALL CM60 - OPPROG segment  
CALL CM70 - DIPROG segment  
CALL CM80 - GBPROG segment  
CALL CM90 - RKPROG segment  
CALL CM100 - AQPROG segment  
CALL CM110 - QOPROG segment  
CALL CM120 - RBPROG segment  
CALL CM140 - RCPROG segment  
CALL CM150 - DRPROG segment

REFERENCES: None

ROUTINE NAME: WGOTO

DESCRIPTION: This routine writes to the file NCMPPL a Fortran-computed GO TO statement based on the information in the argument list.

CALLING SEQUENCE: CALL WGOTO (NSS, NIS, NSN, NAME)

NSS - Starting statement number  
NIS - Number to increment the statement number  
NSN - Number of statement numbers wanted  
NAME - Name of the variable to place on the  
computed GO TO statement

REFERENCED BY: STATEMENT ROUTINE

LOGIC0	WRITNP
LOGIC2	LOGIC2

FILES: NCMPPL - The file that the generated processor Fortran cards are written to

REFERENCES: None

ROUTINE NAME: WMAP

DESCRIPTION: This routine writes the load map directives for the TRASYs processor.

CALLING SEQUENCE: CALL WMAP

REFERENCES:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC2	LOGIC2
	LOGIC3	LOGIC3

FILES: NMAP - File containing processor load map directions

ROUTINE NAME: WPRG0

DESCRIPTION: This routine writes the processor main-program segment to the NCMPPL file.

CALLING SEQUENCE: CALL WPRG0

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC0	WRITNP
	LOGIC2	LOGIC2

FILES: NCMPPL - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPRG2

DESCRIPTION: This routine name is an entry point into the routine WPROG. When this entry point name is called, the Fortran processor subsegment driver RDPROG is written to the file NCMPPL.

CALLING SEQUENCE: CALL WPRG2

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC0	WRITNP
	LOGIC3	LOGIC3

FILES: NCMPPL - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPRG3

DESCRIPTION: This routine name is an entry point into the routine WPROG. When this entry point name is called, the Fortran processor subsegment driver FFPROG is written to the file NCMLP.

CALLING SEQUENCE: CALL WPRG3

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC3	LOGIC3

FILES: NCMLP - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPRG4

DESCRIPTION: This routine name is an entry point into the routine WPROG. When this entry point name is called, the Fortran processor subsegment driver SFPROG is written to the file NCMLP.

CALLING SEQUENCE: CALL WPRG4

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC3	LOGIC3

FILES: NCMLP - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPRG5

DESCRIPTION: This routine name is an entry point into the routine WPROG. When this entry point name is called, the Fortran processor subsegment driver NPPROG is written to the file NCMPL.

CALLING SEQUENCE: CALL WPRG5

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGICO	WRITNP
	LOGIC3	LOGIC3

FILES: NCMPL - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPRG6

DESCRIPTION: This routine name is an entry point into the routine WPROG. When the entry point name is called, the Fortran processor subsegment driver OPPROG is written to the file NCMPL.

CALLING SEQUENCE: CALL WPRG6

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC3	LOGIC3

FILES: NCMPL - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPRG7

DESCRIPTION: This routine name is an entry point into the routine WPROG. When this entry point name is called, the Fortran processor subsegment driver DIPROG is written to the NCMPL file.

CALLING SEQUENCE: CALL WPRG7

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC3	LOGIC3

FILES: NCMPL - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPRG8

DESCRIPTION: This routine name is an entry point into the routine WPROG. When this entry point name is called, the Fortran processor subsegment driver GBPROG is written to the NCML file.

CALLING SEQUENCE: CALL WPRG8

REFERENCED BY: SEGMENT      ROUTINE

LOGIC3      LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPRG9

DESCRIPTION: This routine name is an entry point into the routine WPROG. When this entry point name is called, the Fortran processor subsegment driver RKPROG is written to the NCML file.

CALLING SEQUENCE: CALL WPRG9

REFERENCED BY: SEGMENT      ROUTINE

LOGIC3      LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPRG10

DESCRIPTION: This routine name is an entry point into the routine WPROG. When this entry point name is called, the Fortran processor subsegment driver AQPROG is written to the NCML file.

CALLING SEQUENCE: CALL WPRG10

REFERENCED BY: SEGMENT      ROUTINE

LOGIC3      LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPRG11

DESCRIPTION: This routine name is an entry point into the routine WPROG.  
When this entry point name is called, the Fortran processor  
subsegment driver QOPROG is written to the NCML file.

CALLING SEQUENCE: CALL WPRG11

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are  
written to

ROUTINE NAME: WPRG12

DESCRIPTION: This routine name is an entry point into the routine WPROG.  
When this entry point name is called, the Fortran processor  
subsegment driver RBPROG is written to the NCML file.

CALLING SEQUENCE: CALL WPRG12

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are  
written to

ROUTINE NAME: WPRG13

DESCRIPTION: The routine name is an entry point into the routine WPROG.  
When this entry point name is called, the Fortran processor  
subsegment driver PLPROG is written to the NCML file.

CALLING SEQUENCE: CALL WPRG13

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are  
written to



ROUTINE NAME: WPRG14

DESCRIPTION: This routine name is an entry point into the routine WPROG. When this entry point name is called, the Fortran processor subsegment driver RCPROG is written to the NCML file.

CALLING SEQUENCE: CALL WPRG14

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPRG15

DESCRIPTION: This routine name is an entry point into the routine WPROG. When this entry point name is called, the Fortran processor subsegment driver DRPROG is written to the NCML file.

CALLING SEQUENCE: CALL WPRG15

REFERENCED BY:	<u>SEGMENT</u>	<u>ROUTINE</u>
	LOGIC3	LOGIC3

FILES: NCML - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WPROG

DESCRIPTION: This routine is a collection of entry points that writes the Fortran main routines of the processor subsegments.

CALLING SEQUENCES: WPROG is not called, only the entry points are called

ENTRY POINTS:

CALL WPRG2	-	RDPROG	segment
CALL WPRG3	-	FFPROG	segment
CALL WPRG4	-	SFPROG	segment
CALL WPRG5	-	NPPROG	segment
CALL WPRG6	-	OPPROG	segment
CALL WPRG7	-	DIPROG	segment
CALL WPRG8	-	GBPROG	segment
CALL WPRG9	-	RKPROG	segment
CALL WPRG10	-	AQPROG	segment
CALL WPRG11	-	QOPROG	segment
CALL WPRG12	-	RBPROG	segment
CALL WPRG13	-	PLPROG	segment
CALL WPRG14	-	RCPROG	segment
CALL WPRG15	-	DRPROG	segment

ROUTINE NAME: WRITNP

DESCRIPTION: This routine writes the processor ODPROG subsegment, which overrides the user-input LOGIC DATA block when an error is encountered in the user input. This generated Fortran operation data block causes the node plotter to plot each of the BCS-defined data blocks.

CALLING SEQUENCE: CALL WRITNP (NBCS, NBCSA)

NBCS - Number of BCS-defined surface systems to be plotted  
NBCSA - The array that contains the names of the BCS surface systems

REFERENCED BY: SEGMENT      ROUTINE

LOGICO      LOGICO

FILES: NOUT - System output file  
NCMPL - The file that the generated processor Fortran cards are written to

ROUTINE NAME: WSCOPE (CDC only)

DESCRIPTION: This routine displays a 6-word message, with the last word zeroed, at the computer operator's console. The only information passed to this routine is program status information.

CALLING SEQUENCE: CALL WSCOPE (MSG)

MSG - Message to be written

REFERENCED BY: SEGMENT      ROUTINE

TRASYS      TRASYS  
START      START  
DATARD      DATARD  
LOGICO      LOGICO  
TPGEN      TPGEN

C. FILE DEFINITIONS - PREPROCESSOR

FILE NAME: CMERG

PROGRAM VARIABLE NAME: NCMG

UNIT REFERENCE (UNIVAC/JSC): 12

PURPOSE: This file is the user's card image merge file.

SEGMENT REFERENCE: SEDIT (READ)

FILE NAME: CMPL

PROGRAM VARIABLE NAME: NCMPL

UNIT REFERENCE (UNIVAC/JSC): 20

PURPOSE: This is a compile file that contains the processor FORTRAN routines generated by the preprocessor.

SEGMENT REFERENCES: LOGICO (WRITE)  
LOGIC2 (WRITE)  
LOGIC3 (WRITE)

FILE NAME: DATAI

PROGRAM VARIABLE NAME: NDI

UNIT REFERENCE (UNIVAC/JSC): 4

PURPOSE: This file is generated in the SEDIT segment from the INPUT, EDITI, CMERG, and EMERG files. It contains the users input source data, which are read as the input data file in all other segments of the preprocessor.

SEGMENT REFERENCES: SEDIT (WRITE) DCMNRD (READ)  
GNEDO (READ) FRMFRD (READ)  
QUANRD (READ) SHDWRD (READ)  
ARRYRD (READ) FLUXRD (READ)  
SKIRD (READ) GRSPRD (READ)  
SRFCRD (READ) LOGIC1 (READ)  
BCSRD (READ) LOGIC3 (READ)

FILE NAME: DIR

PROGRAM VARIABLE NAME: NDIR

UNIT REFERENCE (UNIVAC/JSC): 22

PURPOSE: This is the flux data restart file, which contains the flux data request matrix that is generated by the preprocessor and communicated to the processor.

SEGMENT REFERENCES: FLUXRD (WRITE)

FILE NAME: EMERG

PROGRAM VARIABLE NAME: NEMG

UNIT REFERENCE (UNIVAC/JSC): 13

PURPOSE: This file is the user's edit input merge file. It consists of a RSI file, previously generated as an RSO file, mounted and referred to as EMERG.

SEGMENT REFERENCES: MCOLL (WRITE)  
SEDIT (READ)

FILE NAME: FFR

PROGRAM VARIABLE NAME: NFFR

UNIT REFERENCE (UNIVAC/JSC): 21

PURPOSE: This is the form factor restart file that contains the form factor request matrix generated by the preprocessor and communicated to the processor.

SEGMENT REFERENCES: FRMFRD (WRITE)

FILE NAME: GBIRR

PROGRAM VARIABLE NAME: NGBIRR

UNIT REFERENCE (UNIVAC/JSC): 23

PURPOSE: This file contains the correspondence data that were processed in the correspondence data read segment in the preprocessor. These data are communicated to the processor through this file.

SEGMENT REFERENCES: CRSPRD (WRITE)

FILE NAME: INPUT

PROGRAM VARIABLE NAME: NIN

UNIT REFERENCE (UNIVAC/JSC): 5

PURPOSE: This file is the system input file, usually the card reader.

SEGMENT REFERENCES: INITIAL (READ)  
                      MCOLL (READ)  
                      SEDI (READ)

FILE NAME: OUTPUT

PROGRAM VARIABLE NAME: NOUT

UNIT REFERENCE (UNIVAC/JSC): 6

PURPOSE: This file is the system output file, usually the line printers.

SEGMENT REFERENCES: All (WRITE)

FILE NAME: PLSR

PROGRAM VARIABLE NAME: NPLSR

UNIT REFERENCE (UNIVAC/JSC): 25

PURPOSE: This file contains the restart shadow factors that are generated by the preprocessor and passed to the processor.

SEGMENT REFERENCES: SHDWRD (WRITE)

FILE NAME: PUNCH

PROGRAM VARIABLE NAME: NPNCH

UNIT REFERENCE (UNIVAC/JSC): 7

PURPOSE: This file is the system punch file.

SEGMENT REFERENCES: SEDITI (WRITE)

FILE NAME: TAPE1

PROGRAM VARIABLE NAME: NSC1

UNIT REFERENCE (UNIVAC/JSC): 1

PURPOSE: This file is scratch file 1, which is used as temporary storage for intermediate data generated within the preprocessor segments. It is also used to communicate data between segments.

SEGMENT REFERENCES:	QUANRD (WRITE & READ)	FRMFRD (WRITE & READ)
	ARRYRD (WRITE & READ)	LOGIC1 (WRITE)
	SKIRD (WRITE & READ)	LOGIC2 (READ)

FILE NAME: TAPE2

PROGRAM VARIABLE NAME: NSC2

UNIT REFERENCE (UNIVAC/JSC): 2

PURPOSE: This is scratch file 2, which is used as temporary storage for intermediate data generated within the preprocessor segments. It is also used to communicate data between segments.

SEGMENT REFERENCES:	QUANRD (WRITE & READ)
	ARRYRD (WRITE & READ)
	FRMFRD (WRITE & READ)

FILE NAME: TAPE3

PROGRAM VARIABLE NAME: NSC3

UNIT REFERENCE (UNIVAC/JSC): 3

PURPOSE: This is scratch file 3, which is used as temporary storage for intermediate data generated within the preprocessor segments. It is also used to communicate data between segments.

SEGMENT REFERENCES:	FRMFRD (WRITE & READ)
---------------------	-----------------------

FILE NAME: RIO

PROGRAM VARIABLE NAME: NRIO

UNIT REFERENCE (UNIVAC/JSC): 8

PURPOSE: This is the main working random I/O file used in communicating data from one segment to another within the preprocessor, and passing array data, quantities data, and surface data to the processor.

SEGMENT REFERENCES:   INITAL (WRITE)   SDPSS2 (WRITE)  
                      QUANRD (WRITE)   BCSRD (WRITE)  
                      ARRAYD (WRITE)   LOGIC2 (WRITE & READ)

FILE NAME: RIO2

PROGRAM VARIABLE NAME: NRIO2

UNIT REFERENCE (UNIVAC/JSC): 10

PURPOSE: This file is used to store data generated by SRFCRD for use by SDPSS2.

SEGMENT REFERENCES:   SRFCRD (WRITE)   SDTPSS2 (READ)

FILE NAME: RIOS

PROGRAM VARIABLE NAME: NRIOS

UNIT REFERENCE (UNIVAC/JSC): 9

PURPOSE: This file is the scratch random I/O file. It is used as temporary storage for intermediate data generated within the preprocessor and is also used to communicate data between segments.

SEGMENT REFERENCES:   SRFCRD (WRITE)  
                      SDPSS2 (READ)

FILE NAME: RSI

PROGRAM VARIABLE NAME: NRSI

UNIT REFERENCE (UNIVAC/JSC): 14

PURPOSE: This file is the master restart input file.

SEGMENT REFERENCES: EDIT

FILE NAME: RSO

PROGRAM VARIABLE NAME: NRSO

UNIT REFERENCE (UNIVAC/JSC): 15

PURPOSE: This file is the master restart output file.

SEGMENT REFERENCES: START

FILE NAME: SQNTL

PROGRAM VARIABLE NAME: NSQNTL

UNIT REFERENCE (UNIVAC/JSC): 16

PURPOSE: This is the sequential communication file between the  
preprocessor and processor.

SEGMENT REFERENCES: TPGEN (WRITE)

FILE NAME: MAP

PROGRAM VARIABLE NAME: NMAP

UNIT REFERENCE (UNIVAC/JSC): 27

PURPOSE: This file contains the processor map directives.

SEGMENT REFERENCES: LOGIC2, LOGIC3



D. VARIABLE DEFINITIONS - PREPROCESSOR

LABELED COMMON /CARD/

NCDTYP - Variable that contains the first 6 characters of the last input card read

IND - Array variable that contains the 66 characters from Col. 7 thru Col. 72 of the last input card read

NNED - New edit number of the last card read

NOED - Old edit number of the last card read

LMOD - Modification label of the last card read

ID - Number of decoded words in the NDOT and IDOT arrays

NDOT - The array containing the decoded words of the last input card that was decoded

IDOT - The array containing the type of word decoded in the NDOT array

= -(IRI) Integer word

= -(IRF) Floating-point word

= -(IRS) Special character

= + NUMBER Hollerith word character count

JDOT - The array containing the start Col. numbers of the input card data fields

IBE - Last word in the INB buffer array that has been processed

IBT - Last word in the INB buffer array filled with input cards

INB - Buffer array that the 46 cards are read into

IEOF - End of file flag

= 0 No end of file encountered

= 1 End of file has been encountered

NER - Subroutine independent error flag

NRECN - The record number currently being processed on file DTI

LABELED COMMON /CHRCR/

N1RA = 1RA  
 N1RB = 1RB  
 N2RC = 1RC  
 N1RD = 1RD  
 N1RE = 1RE  
 N1RF = 1RF  
 N1RG = 1RG  
 N1RH = 1RH  
 N1RI = 1RI  
 N1RK = 1RK  
 N1RL = 1RL  
 N1RM = 1RM  
 N1RN = 1RN  
 N1RO = 1RO  
 R1RP = 1RP  
 N1RQ = 1RQ  
 N1RR = 1RR  
 N1RS = 1RS  
 N1RT = 1RT  
 N1RU = 1RU  
 N1RV = 1RV  
 N1RW = 1RW  
 N1RX = 1RX  
 N1RY = 1RY  
 N1RZ = 1RZ  
 N1RO = 1RO  
 N1R9 = 1R9  
 N1RPR = 1R.  
 N1RDO = 1R\$  
 N1RAS = 1R\*  
 N1RSL = 1R/  
 N1RPL = 1R+  
 N1RMI = 1R-  
 N1RCM = 1R,  
 N1REQ = 1R=  
 N1RLP = 1R(  
 N1RRP = 1R)  
 N1RBK = 1R

LABELED COMMON /CONST/

NPAGE - Page number of the current page being written to the  
output file  
NLINE - Line number of the last line written to the output file  
MLINE - Maximum number of lines allowed on an output page less  
3 lines  
MR77 - Variable used in masking operations  
MR67 - Variable used in masking operations  
MR127 - Variable used in masking operations  
MRBLK - Variable used in checking for blank characters  
NBLANK - A word containing an all-blank Hollerith code  
MAXBC - Maximum length of blank common  
MAXF - Maximum core field length possible for the run  
NA - Contains the Hollerith code (N/A)  
ISPCL - Denotes a special character decoded (-1RS)  
INTEG - Denotes an integer word decoded (-1RI)  
IFLT - Denotes a floating-point number decoded (-1RF)  
MLABEL - Current edit modifier label  
NONE - Contains the Hollerith code (none)  
INPTSF - Multi-shadow factor input block flag  
IEQFF - Equivalent form factor flag  
FFZERO - Zero form factor flag  
FFNAC - Flag to indicate if node array encountered in form factor  
data  
MR47 - Variable used in masking operations

# LABELED COMMON /CPCORE/

TCPST - Central processor (CP) time of start of preprocessor  
ICRMC - Minimum core for model collect editing  
TCPE - CP time for source editing  
ICRE - Minimum core for source editing  
TCPQ - CP time for compiling quantities data  
ICRQ - Minimum core for compiling quantities data  
TCPA - CP time for compiling array data  
ICRA - Minimum core for compiling array data  
TCPS1 - CP time for compiling surface data in pass 1  
ICRS1 - Minimum core for compiling surface data in pass 1  
TCPS2 - CP time for compiling surface data in pass 2  
ICRS2 - Minimum core for compiling surface data in pass 2  
TCPB - CP time for compiling BCS data  
ICRB - Minimum core for compiling BCS data  
TCPFF - CP time for compiling form factor data  
ICRFF - Minimum core for compiling form factor data  
TCPSW - CP time for compiling shadow data  
ICRSW - Minimum core for compiling shadow data  
TCPF - CP time for compiling flux data  
ICRF - Minimum core for compiling flux data  
TCPC - CP time for compiling correspondence data  
ICRC - Minimum core for compiling correspondence data  
TCPO - CP time for compiling operation data  
ICRO - Minimum core for compiling operation data  
TCPSR - CP time for initialization - RSTRTO and SQNTL  
ICRSR - Minimum core initialization - RSTRTO and SQNTL  
TCPD - CP time for processing documentation data block  
ICRD - Minimum core for processing documentation data  
TCPS - CP time for preprocessing subroutine data  
ICRS - Minimum core for compiling subroutine data  
TCPR - CP time for preprocessing restart data  
ICRR - Minimum core for processing restart data  
IPPST - Preprocessor (PP) time of start of preprocessor  
IPPIN - PP time for initialization  
IPPE - PP time for source editing  
IPPQ - PP time for compiling quantities data  
IPPA - PP time for compiling array data  
IPPS1 - PP Time for compiling surface data in pass 1  
IPPS2 - PP time for compiling surface data in pass 2  
IPPB - PP time for compiling BCS data  
IPPPF - PP time for compiling form factor data  
IPPSW - PP time for compiling shadow data  
IPFF - PP time for compiling flux data  
IPPC - PP time for compiling correspondence data  
IPPO - PP time for compiling operations data  
IPFSR - PP time for initialization of SQNTL and RSTRTO  
IPPD - PP time for documentation data block

LABELED COMMON /CPCORE/ (continued)

IPPO - PP time for processing documentation data block  
IPPS - PP time for compiling subroutine data  
IPPR - PP time for compiling restart data  
ICPIN - CP time for initialization  
ICRIN - Minimum core for initialization

LABELED COMMON /CVCONS/

NBITS - Maximum number of bits in a computer word  
NBITC - Maximum number of bits in a Hollerith character  
NW1T72 - Minimum number of computer words to read in Col. 1 thru  
Col. 72 of an input card  
NW7T72 - Minimum number of computer words to read in Col. 7 thru  
Col. 72 of an input card  
NCHAR - Number of Hollerith characters in a computer word

LABELED COMMON /CVFAC/

DTR - Degrees to radians  
RTD - Radians to degrees  
PI - PI

LABELED COMMON /DIMES/

NSRFC - Number of surfaces read in by the preprocessor  
NWDSMX - Maximum number of words on longest surface record  
NOFBCS - Number of BCS read in by the preprocessor  
NTSPSF - Total number of specular surfaces

LABELED COMMON /ERROR/

NERR - Total number of errors detected by the preprocessor  
NCTN - Total number of cautions detected by the preprocessor  
KERR - Array of line numbers where errors exist  
KCTN - Array of line numbers where cautions exist  
IERR - Number of error words filed in the KERR array  
ICTN - Number of caution words filed in the KCTN array

LABELED COMMON /FILE/

NSC1 - Scratch sequential file 1  
NSC2 - Scratch sequential file 2  
NSC3 - Scratch sequential file 3  
NDI - Output of source editor and input to preprocessor  
NIN - Card reader input file  
NOUT - Printer output file  
NRIO - Main working random I/O file  
NRIOS - Scratch random I/O file  
NCMG - Card input merge file  
NEMG - Edit input merge file  
NRSI - Restart file input (permanent)  
NRSO - Restart file output (permanent)  
NSQNTL - Sequential communication file  
NRTI - Restart file input (temporary)  
NCMPL - Generated FORTRAN Program output file  
NFFR - Form factor restart file  
NDIR - Flux data restart file  
NGBIRR - Correspondence data file  
NPLSR - Shadow factor restart file  
NTQR - Not used  
NUSER1 - User file 1  
NUSER2 - User file 2  
NFF - Not used  
NDILB - Not used  
NGBIR - Not used  
NGBSO - Not used  
NPLS - Shadow factor file for processor  
NTQ - Not used  
NRTO - Restart file output (temporary)  
NBCDOU - MITAS-SINDA BCD interface tape  
NPNCH - System punch output file  
NRIO2 - Working random I/O file  
NINFO - User information file

LABELED COMMON /FILET/

KRSI     = 14  
KRSO     = 15  
KRTI     = 17  
KRTO     = 18  
KCMG     = 12  
KEMG     = 13  
KBCDOU   = 29  
KTRAJ     = 28

LABELED COMMON /FNAME/

These variables are used to transfer file information from the options card read-in routine to the edit and history routines

JCMG     = 1H  
JEMG     = 1H  
JRSI     = 1H  
JRSO     = 1H  
JRTI     = 1H  
JBCDOU   = 1H  
JTRAJ     = 1H  
JUSER1   = 1H  
JUSER2   = 1H

LABELED COMMON /HEAD/

LMDTE     - Date of last modification  
LWRSN     - Current TRASYS version and modification number  
NDTE     - Job run data  
NTME     - Job run start time  
NJOB     - Job number  
NTITLE    - Array containing the TRASYS internal title that is printed at the top of each output page  
ITITLE    - Array containing the user-input title for the primary model. This title is printed at the top of each printed page

LABELED COMMON /IDSTR/

IDSTR     - Array containing guiding directives for all information on the files created by the input data blocks.



LABELED COMMON /OPTION/

IRSTRT - Flag for designating a restart or start run  
IPRNTS - Flag for type of source edit printing  
IPNCHS - Flag for type of source edit punching  
IGO - Flag for editing and preprocessing the data, or just  
editing the data and terminating  
IPRNTE - Flag for type of edit directive output  
IRLBLS - Flag for relabeling the source  
MDLNM - Primary model name that is to be processed  
MDLNNM - Input model name to be changed to MDLNNM after editing  
IDMPD - Flag for printing documentation data block  
IERCNT - Preprocessor error - normal continuation flag  
MAXFL - Maximum field length for processor execution  
NOINP - Not used  
IPINFO - Flag for printing information data file  
ERPLOT - Generate node plot in case of errors

LABELED COMMON /RNDMIO/

MRREC - Maximum number of records that can be written to the  
random I/O file, RIO  
NRREC - Number of records that have been written to the random  
I/O file, RIO  
IRINDX - Not used  
MRREC2 - Maximum number of records that can be written to random  
I/O file, RIO2  
NRREC2 - Number of records that have been written to random I/O  
file, RIO2  
JRINDX - Not used

LABELED COMMON /RNDMRC/

NLCQV - Number of words in the control constant value array record  
NRUQD - Random I/O record that the user constant directory array record is written to  
NLUQD - Number of words in the user constant directory array record  
NLUQV - Number of words in the user constant value array record  
NRAND - Random I/O record that the user array name directory array is written to  
NRPAD - Random I/O record that the user array position directory array is written to  
NLAD - Number of words in the user array directory array record  
NLAV - Number of words in the user array value array record  
NRIS - Random I/O record that the surface index is written to  
NLIS - Number of words in the surface index record  
NLIN - Number of words in the node index record  
NRBCSD - Random I/O record that the BCS directory is written to  
NLBCSD - Number of words in the BCS directory record  
NLBCSR - Number of words in the BCS index record  
NLSD - Number of words in the step directory  
NLBCSN - Number of BCS names written to the NRBCSN record

LABELED COMMON /TRACE/

ITRC12 - Trace printout flag for segment MCOLL  
ITRCED - Trace printout flag for segments SEDIT & GNEDO  
ITRCDM - Not used  
ITRC21 - Trace printout flag for segment QUANRD  
ITRC22 - Trace printout flag for segment ARRYRD  
ITRC23 - Trace printout flag for segment SKIRD  
ITRC24 - Trace printout flag for segment SRFCRD  
ITRC25 - Trace printout flag for segment SDPSS2  
ITRC26 - Trace printout flag for segment BCSR  
ITRC27 - Not used  
ITRCFF - Trace printout flag for segment FRMFRD  
ITRC29 - Trace printout flag for segments FLUXRD and CRSRD  
ITRC31 - Trace printout flag for segment LOGIC1  
ITRC32 - Trace printout flag for segment LOGIC2  
ITRC33 - Trace printout flag for segment LOGIC3  
ITRCTG - Trace printout flag for segment TPGEN  
ITRCRS - Trace printout flag for segment RESTRT  
ITRCON - Trace flag to be checked for  
ITRCSF - Trace printout flag for segment SHDWRD

E. ENTRY POINTS - PREPROCESSOR

<u>ENTRY POINT</u>	<u>SUBROUTINE</u>
CHECTP	CHEC
INRDB	INRDD
INRDDC	INRDD
INRDO	INRDD
INRDOD	INRDD
KPIE	KPI
WAQCOM	WCOM
WCMCOM	WCOM
WDICOM	WCOM
WDRCOM	WCOM
WFFCOM	WCOM
WGBCOM	WCOM
WMFCOM	WCOM
WNPCOM	WCOM
WOPCOM	WCOM
WPLCOM	WCOM
WQOCOM	WCOM
WRBCOM	WCOM
WRCCOM	WCOM
WSFCOM	WCOM
WAQSEG	WPROG
WCMSEG	WPROG
WDISEG	WPROG
WDRSEG	WPROG
WFFSEG	WPROG
WGBSEG	WPROG
WMFSEG	WPROG
WNPSEG	WPROG
WOCSEG	WPROG
WOPSEG	WPROG
WPLSEG	WPROG
WQOSEG	WPROG
WRBSEG	WPROG
WRCSEG	WPROG
WRDSEG	WPROG
WSFSEG	WPROG

### III. PART 2 - PROCESSOR LIBRARY

---

#### A. SEGMENT DEFINITIONS

SEGMENT NAME: AQCAL

PURPOSE: This segment computes absorbed heat rates in two wavebands, accounting for specular and diffuse reflection (Fig. III-1).

RESTRICTIONS: Appropriate direct-irradiation and gray-body factors must be located on the DI, GBIR, and GBSO files before execution.

CALLING SEQUENCE: L AQCAL

OUTPUT: Absorbed total heat rates written to file TQ.

SEGMENT NAME: CMCAL

PURPOSE: This segment combines form factors according to correspondence data, resulting in a form factor matrix of reduced size.

RESTRICTIONS: Requires form factors on file FF.

CALLING SEQUENCE: L CMCAL

OUTPUT: Writes combined form factors to file FF.

SEGMENT NAME: DICAL

PURPOSE: This segment computes solar, planetary, and albedo irradiation incident on spacecraft nodes (Fig. III-2).

RESTRICTIONS: Execution is possible after previous calls have been made to define spacecraft geometry, location in space, characteristics and distances of heat source bodies, and computation accuracy parameters.

CALLING SEQUENCE: L DICAL

OUTPUT: Printed, punched, written to files DI, RTO, and PLS.

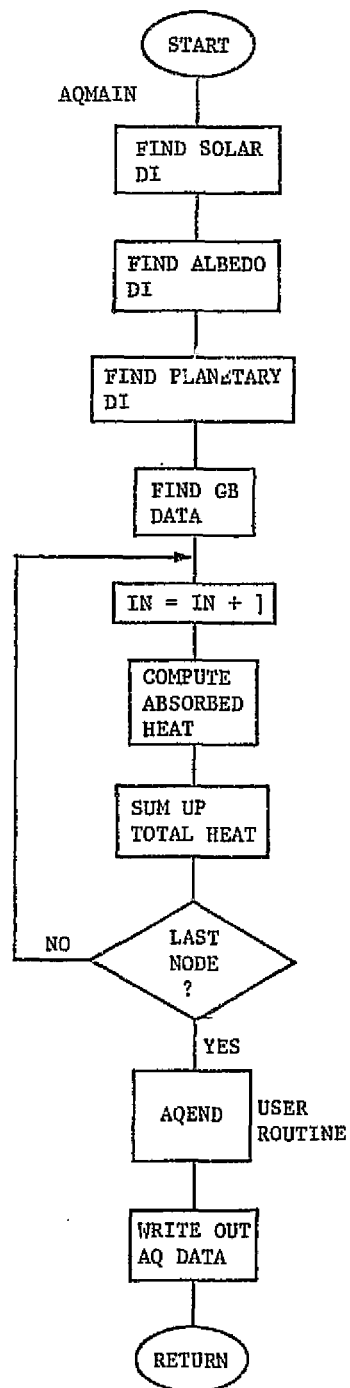


Figure III-1 Segment AQCAL Flow Diagram

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

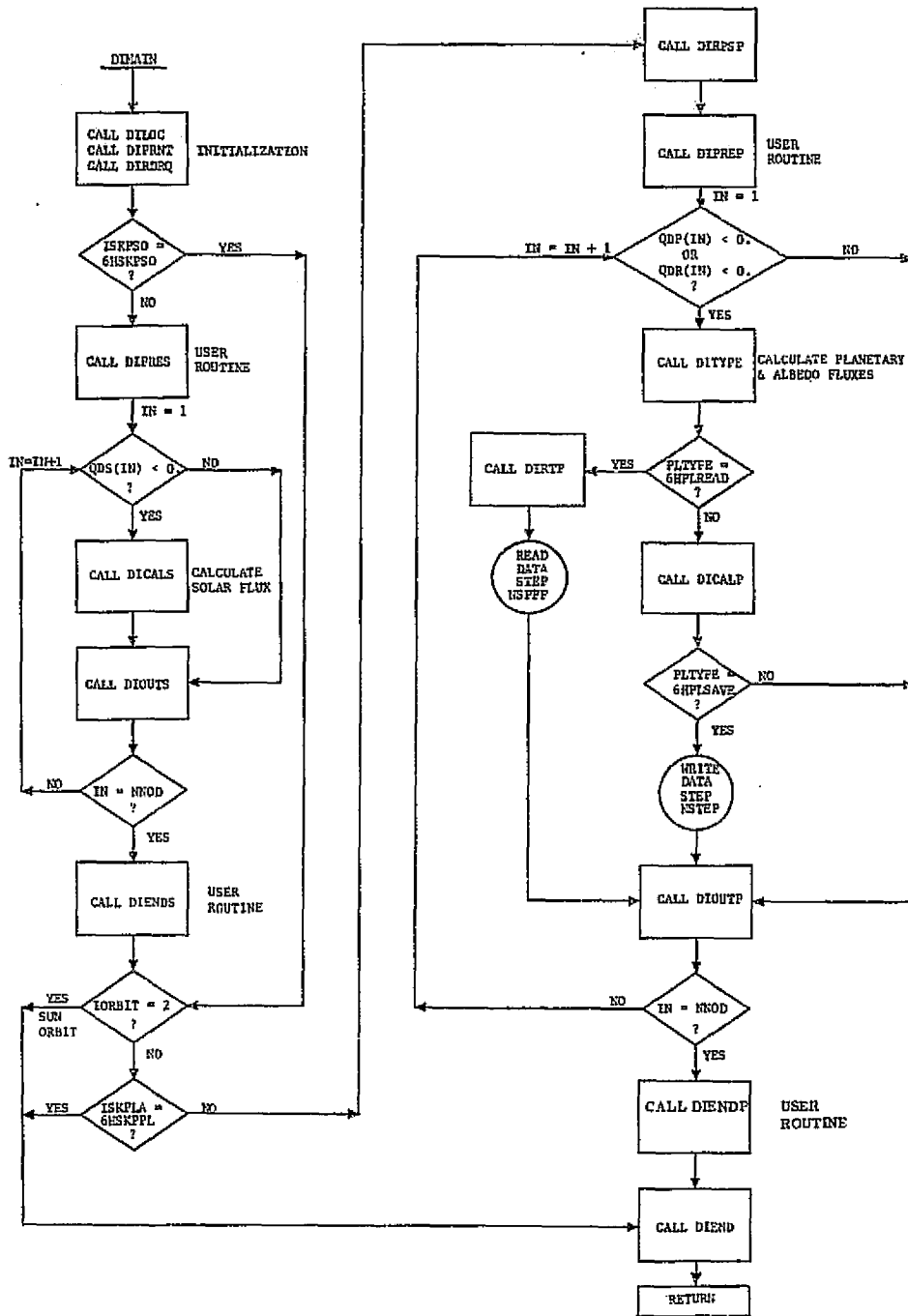


Figure III-2 Segment DICAL Flow Diagram

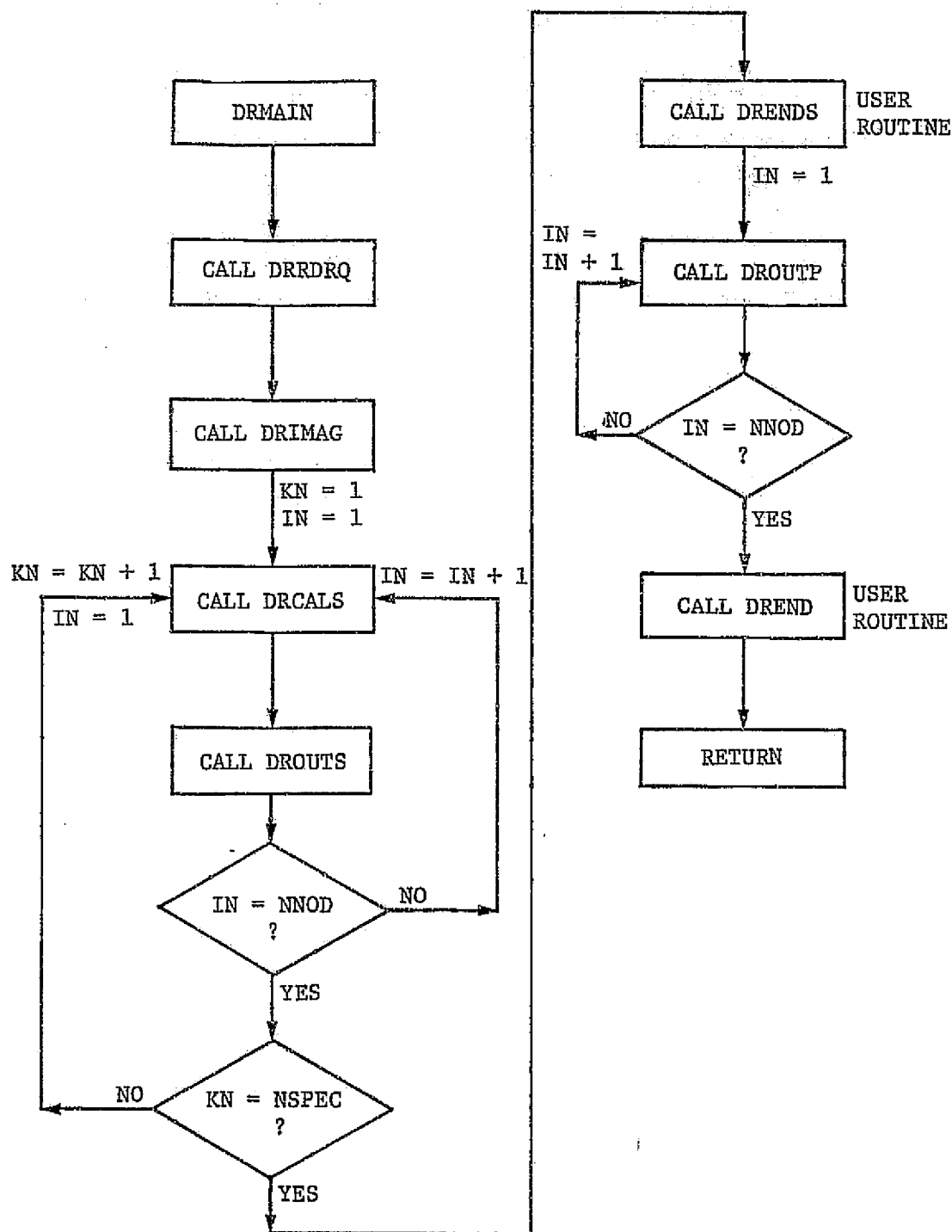


Figure III-3 Segment DRCAL Flow Diagram

SEGMENT NAME: FFCAL

PURPOSE: This segment calculates all form factors for the active configuration (Fig. III-4).

RESTRICTIONS: None

CALLING SEQUENCE: L FFCAL

OUTPUT: Printed, punched, written to file FF.

SEGMENT NAME: DRCAL

PURPOSE: This segment computes the components of direct irradiation that reach the nodes via specular reflections in other nodes.

RESTRICTIONS: Requires direct irradiation data on file DI.

CALLING SEQUENCE: L DRCAL

OUTPUT: Writes direct irradiation that includes specular reflection components to file DI. Print file output.

SEGMENT NAME: RBCAL

PURPOSE: This segment computes the components of internode form factors that result from images of nodes in specular surfaces.

RESTRICTIONS: Requires direct form factor data on file FF.

CALLING SEQUENCE: L RBCAL

OUTPUT: Writes modified form factors, containing specular components, to the FF file.



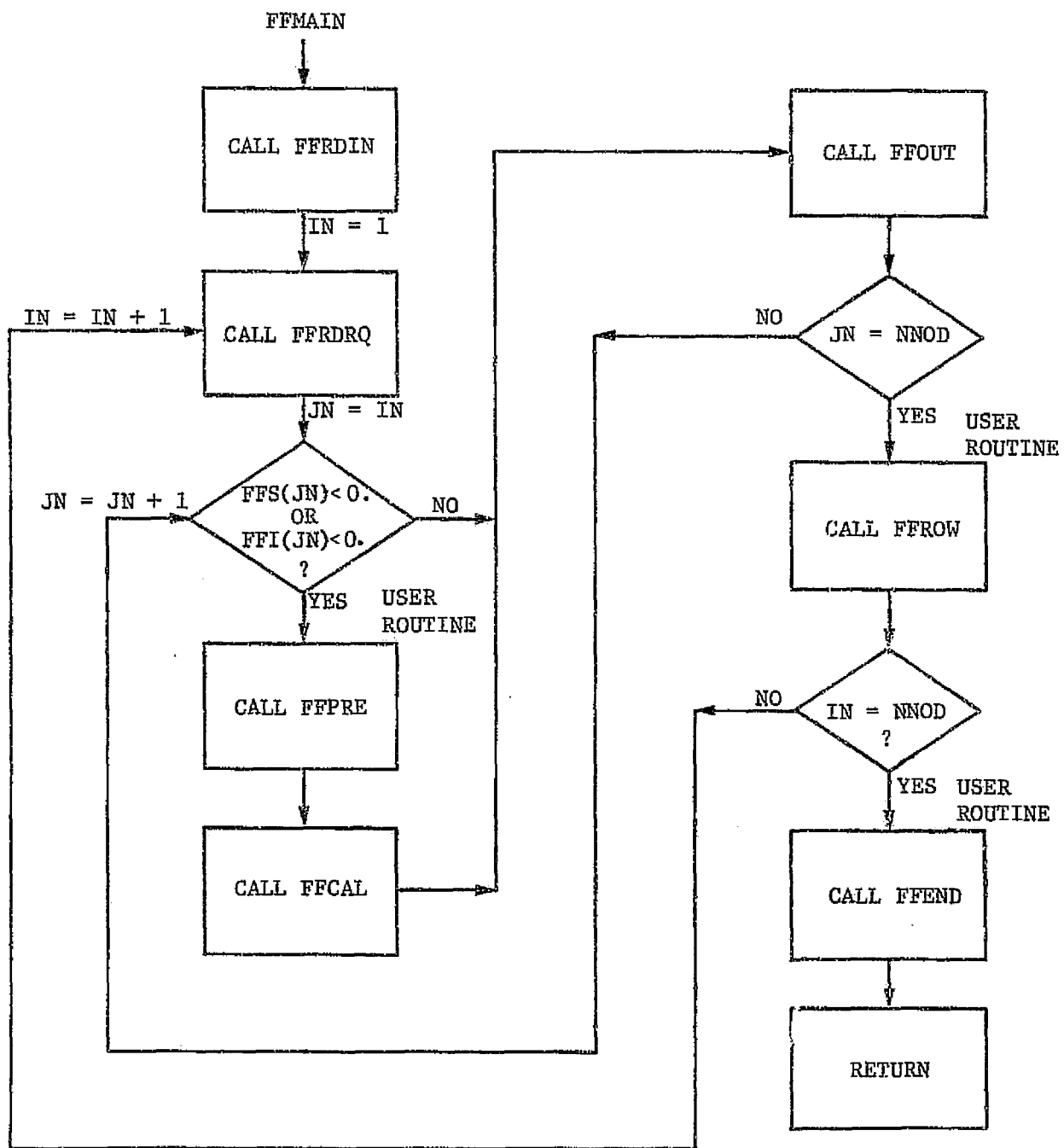


Figure III-4 Segment FFCAL Flow Diagram

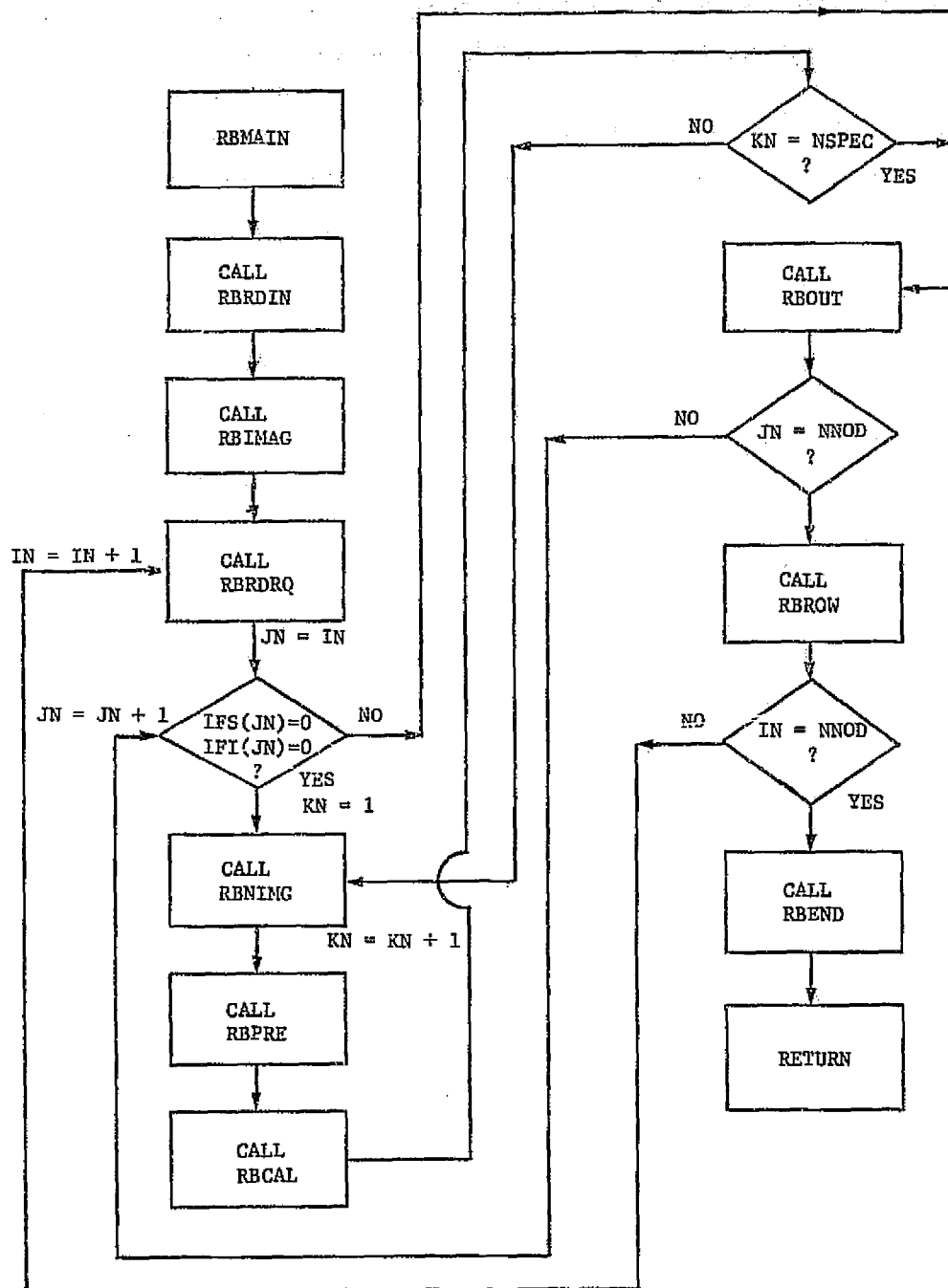


Figure III-5 Segment RBCAL Flow Diagram

SEGMENT NAME: GBCAL

PURPOSE: This segment computes and stores the gray-body factor matrix (Fig. III-6).

RESTRICTIONS: Requires form factor data on file FF.

CALLING SEQUENCE: L      GBCAL

OUTPUT: Writes gray-body factor matrices to GBIR and/or GBSO.

SEGMENT NAME: NPLOT

PURPOSE: This segment generates pictorial plots of nodal surfaces (Fig. III-7).

RESTRICTIONS: None

CALLING SEQUENCE: L      NPLOT

OUTPUT: Printed, plot file.

SEGMENT NAME: OPLOT

PURPOSE: This segment generates pictorial plots of the spacecraft in orbit (Fig. III-8).

RESTRICTIONS: None

CALLING SEQUENCE: L      OPLOT

OUTPUT: Printed, plot file.

SEGMENT NAME: PLOT

PURPOSE: This segment generates function vs time plots of absorbed and incident heat rates and fluxes. When used in conjunction with operations block FORTRAN that writes data to a plot data unit, this segment provides a general x vs y plot capability (Fig. III-9).

RESTRICTIONS: Reference Subroutine PLDATA

CALLING SEQUENCE: L      PLOT

OUTPUT: Printed, plot file.

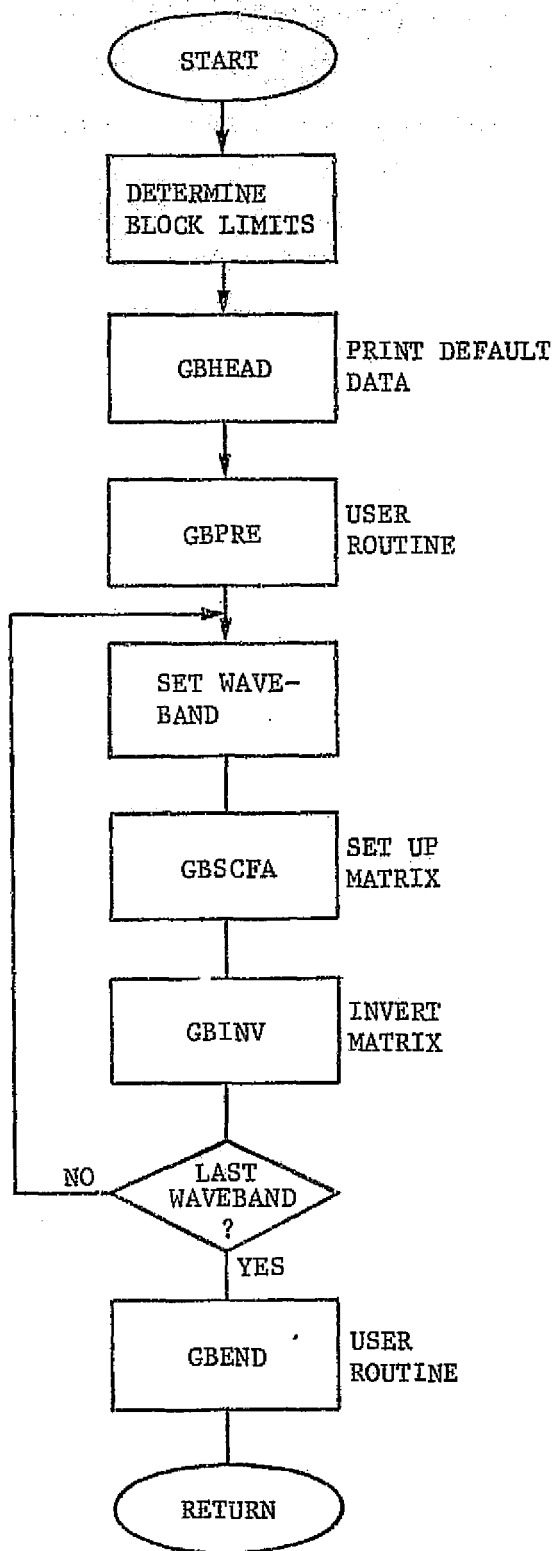


Figure III-6 Segment GBCAL Flow Diagram

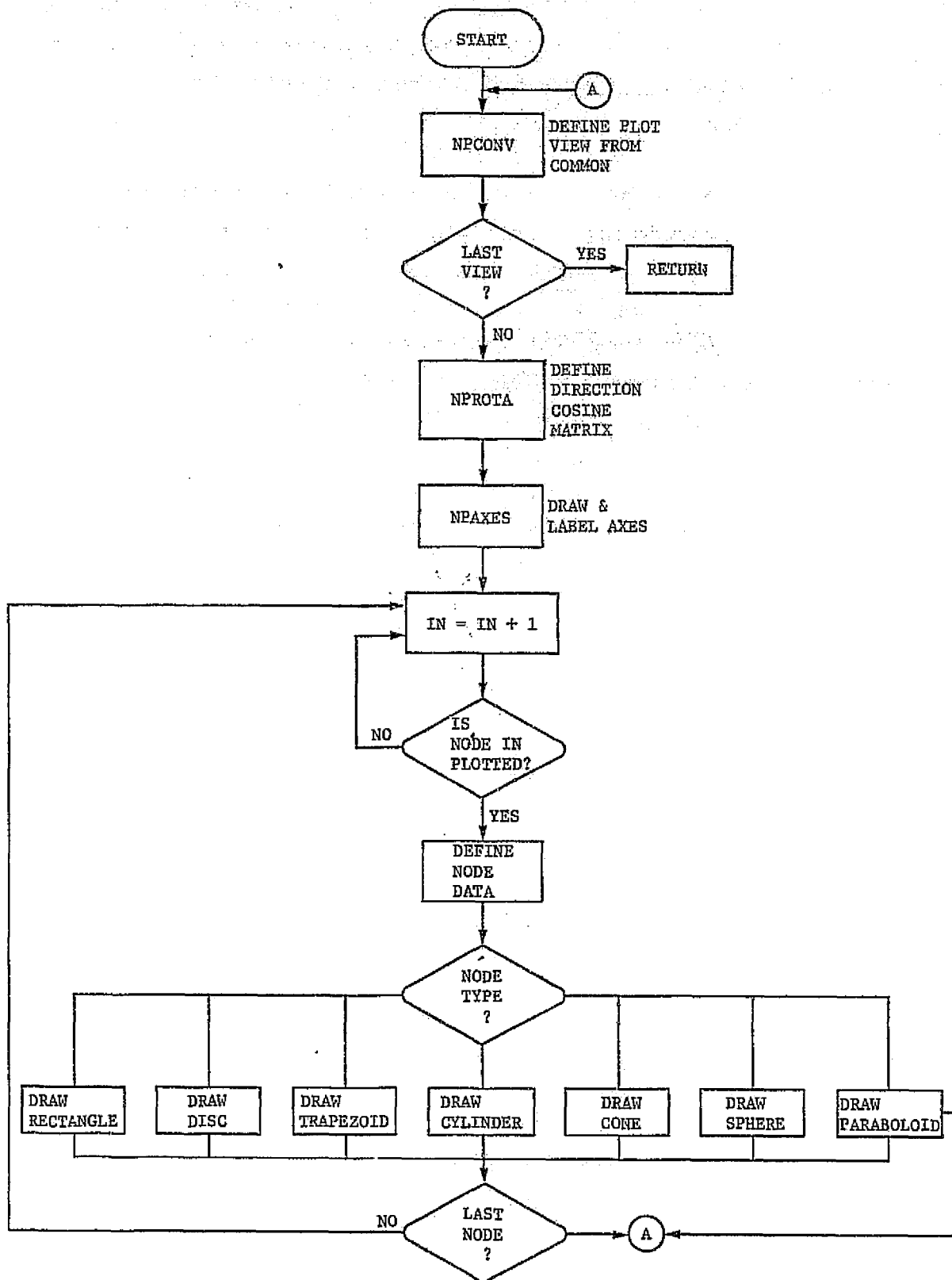


Figure III-7 Segment NPLLOT Flow Diagram

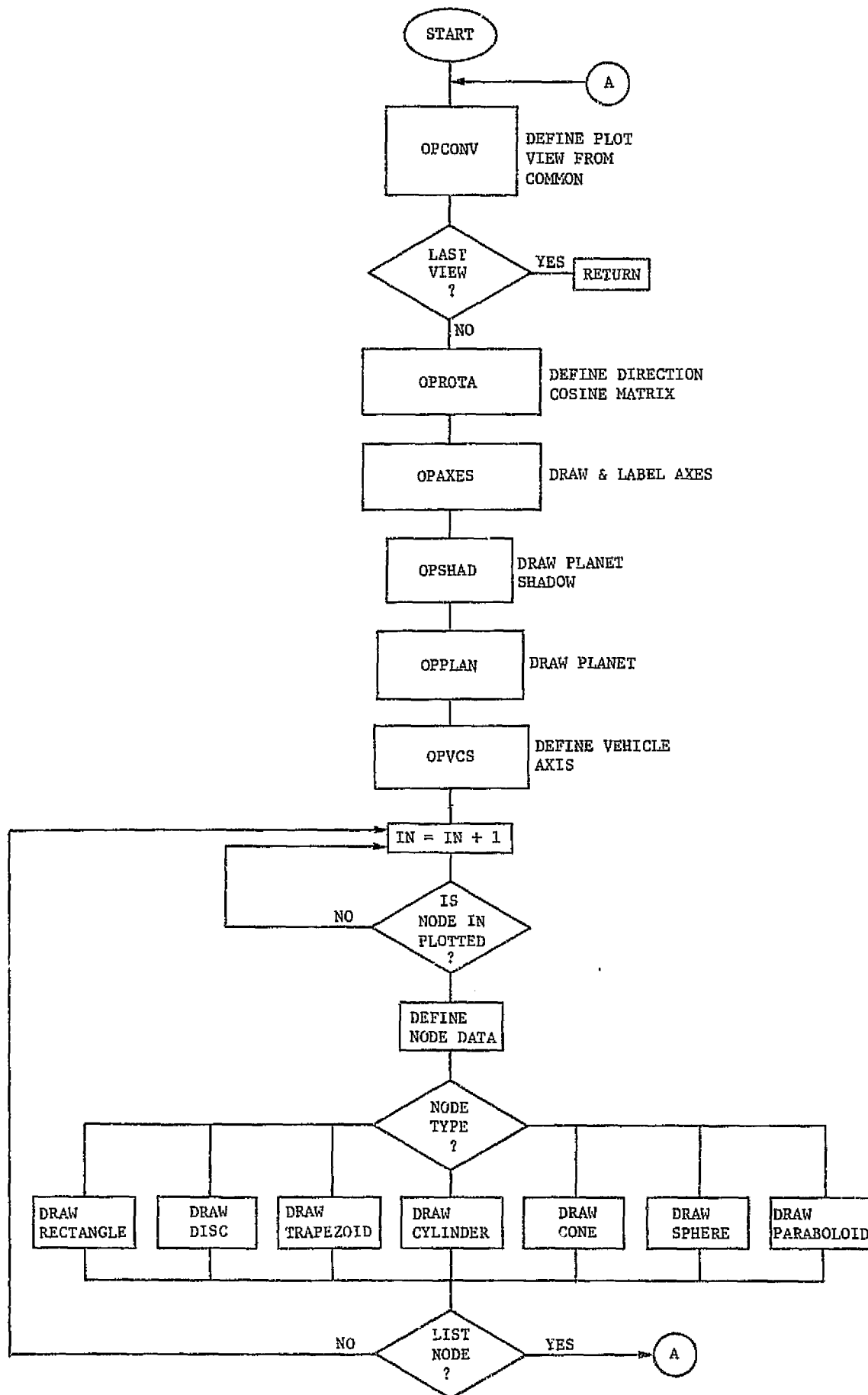


Figure III-8 Segment OPLLOT Flow Diagram

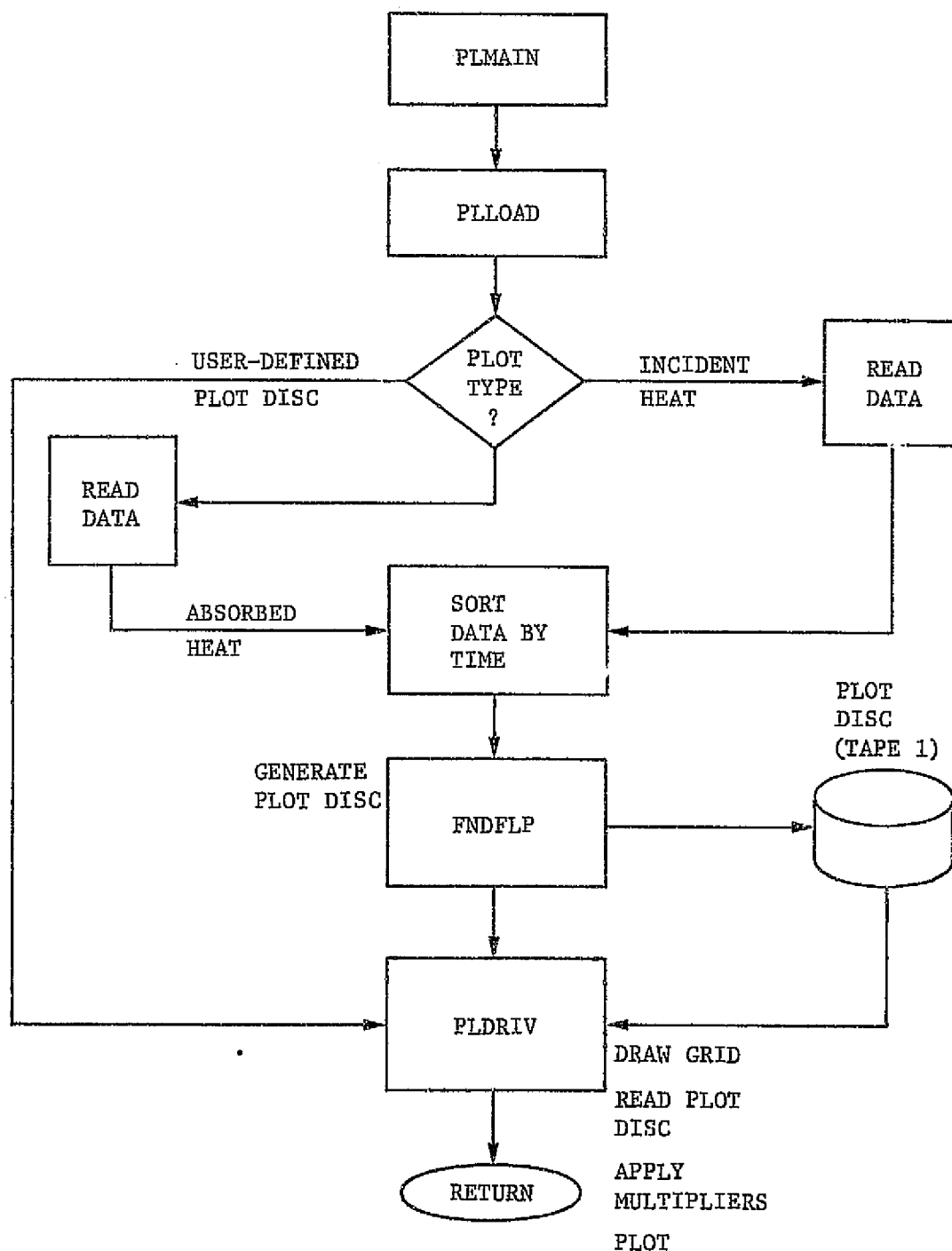


Figure III-9 Segment PLOT Flow Diagram

SEGMENT NAME: QOCAL

PURPOSE: This segment accesses absorbed flux data and generates orbital average and absorbed flux vs time arrays (Fig. III-10).

RESTRICTIONS: Requires data on the TQ file.

CALLING SEQUENCE: L QOCAL

OUTPUT: Printed, punched, BCD tape (File BCDOU).

SEGMENT NAME: RCCAL

PURPOSE: This segment computes radiation conductors and simplifies and condenses these conductors using the ERN and MESS techniques (Fig. III-11).

RESTRICTIONS: Requires data on the GBIR file.

CALLING SEQUENCE: L RCCAL

OUTPUT: Printed, punched, BCD tape (file BCDOU).

SEGMENT NAME: RKCAL

PURPOSE: This segment computes radiation conductor values (Fig. III-12).

RESTRICTIONS: Requires data on the GBIR file.

CALLING SEQUENCE: L RKCAL

OUTPUT: Printed, punched, BCD tape (file BCDOU).

SEGMENT NAME: SFCAL

PURPOSE: This segment computes tables of internode blockage (shadow) factors for use in direct irradiation calculations. When a complete shadow factor tape is supplied SFCAL is executed to pass the shadow tables to file PLSR and initialize DICAL to compute irradianations using the shadow tables (Fig. III-13).

RESTRICTIONS: None

CALLING SEQUENCE: L SFCAL

OUTPUT: Printed, binary tape (files NPLS and RSO).



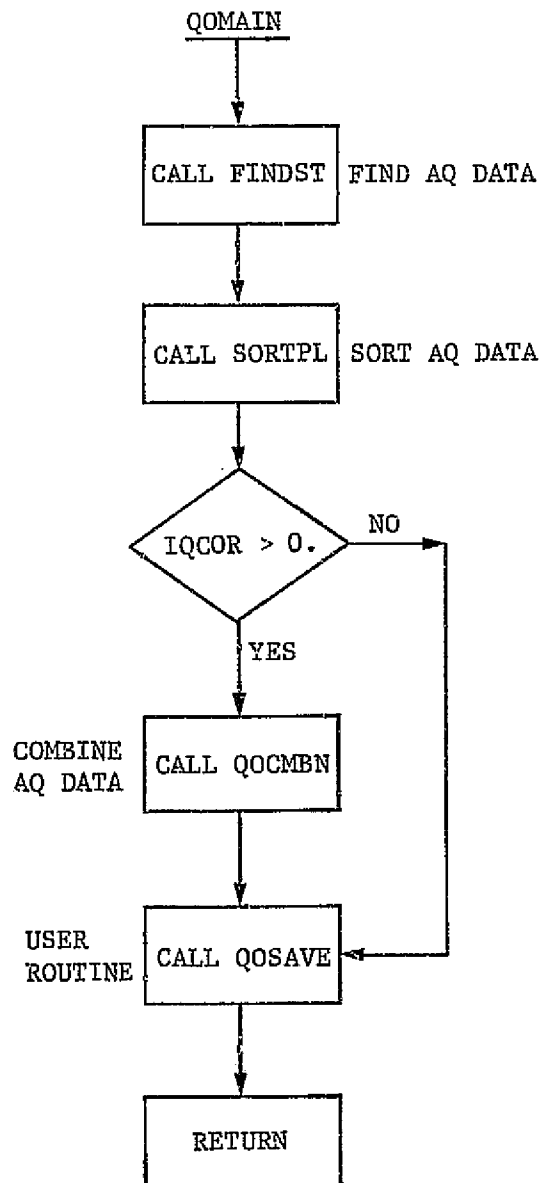


Figure III-10 Segment QOCAL Flow Diagram

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

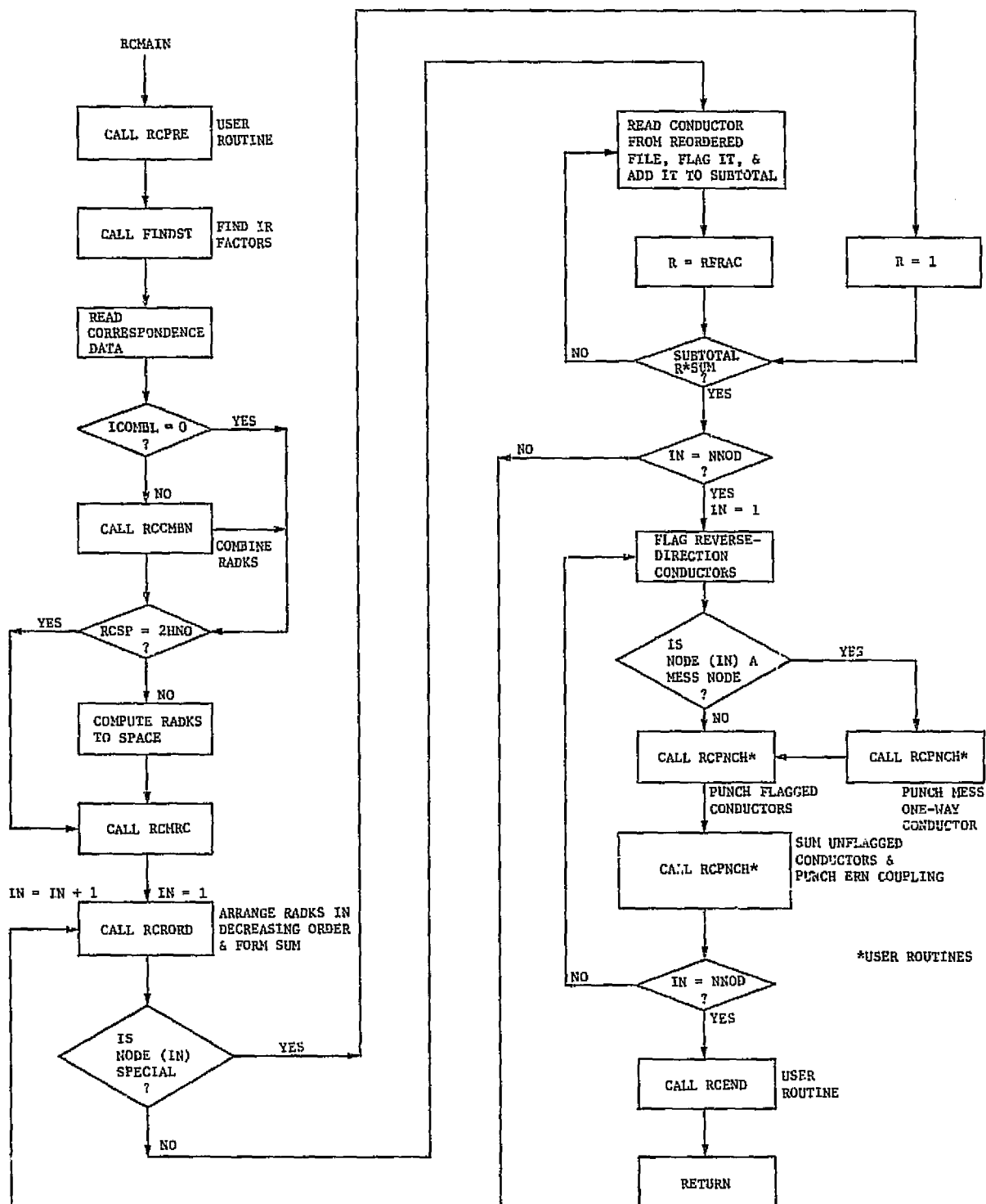


Figure III-11 Segment RCCAL Flow Diagram

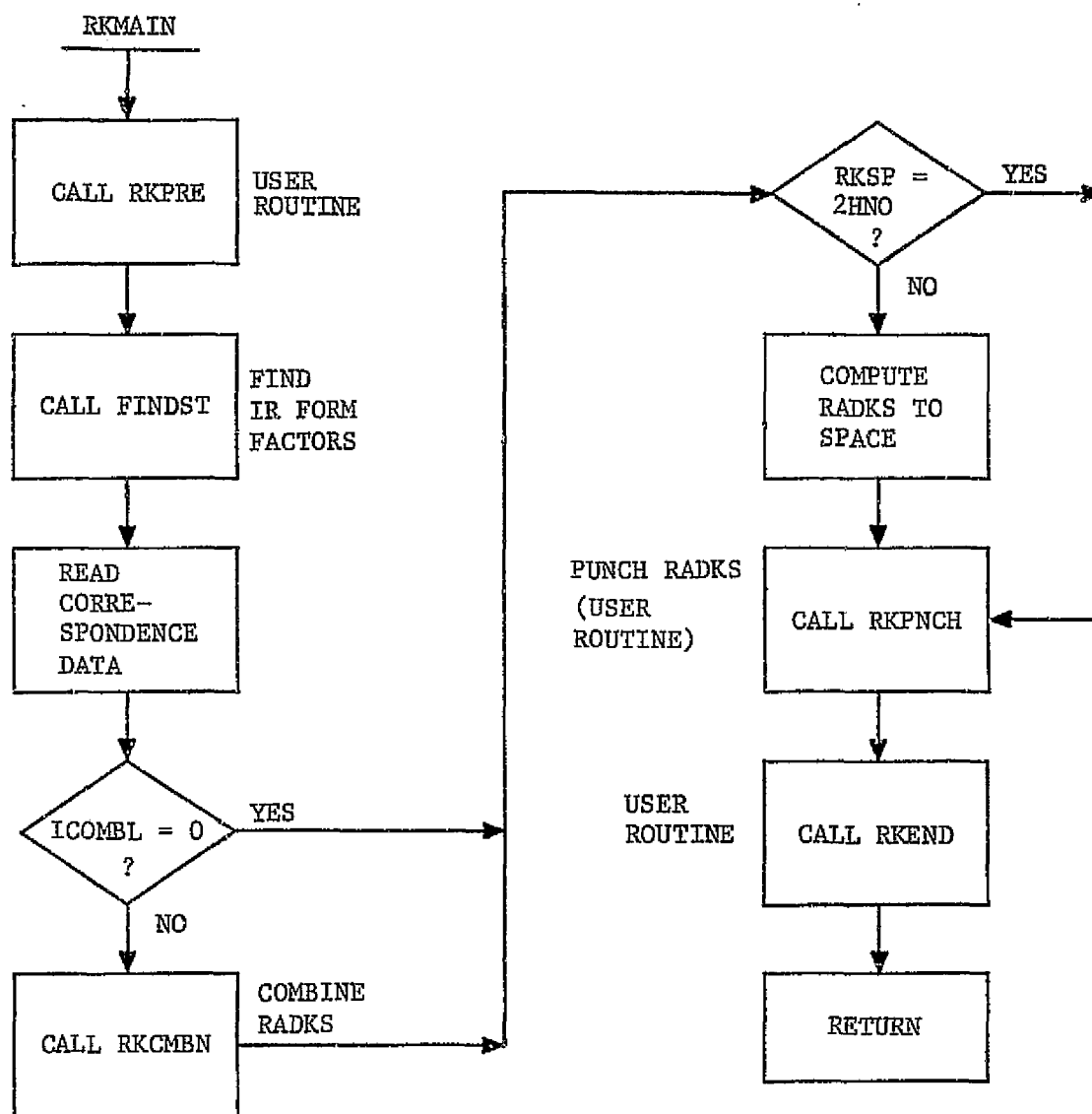


Figure III-12 Segment RKCAL Flow Diagram

REPRODUCIBILITY OF THE  
ORIGINAL PAGE IS POOR

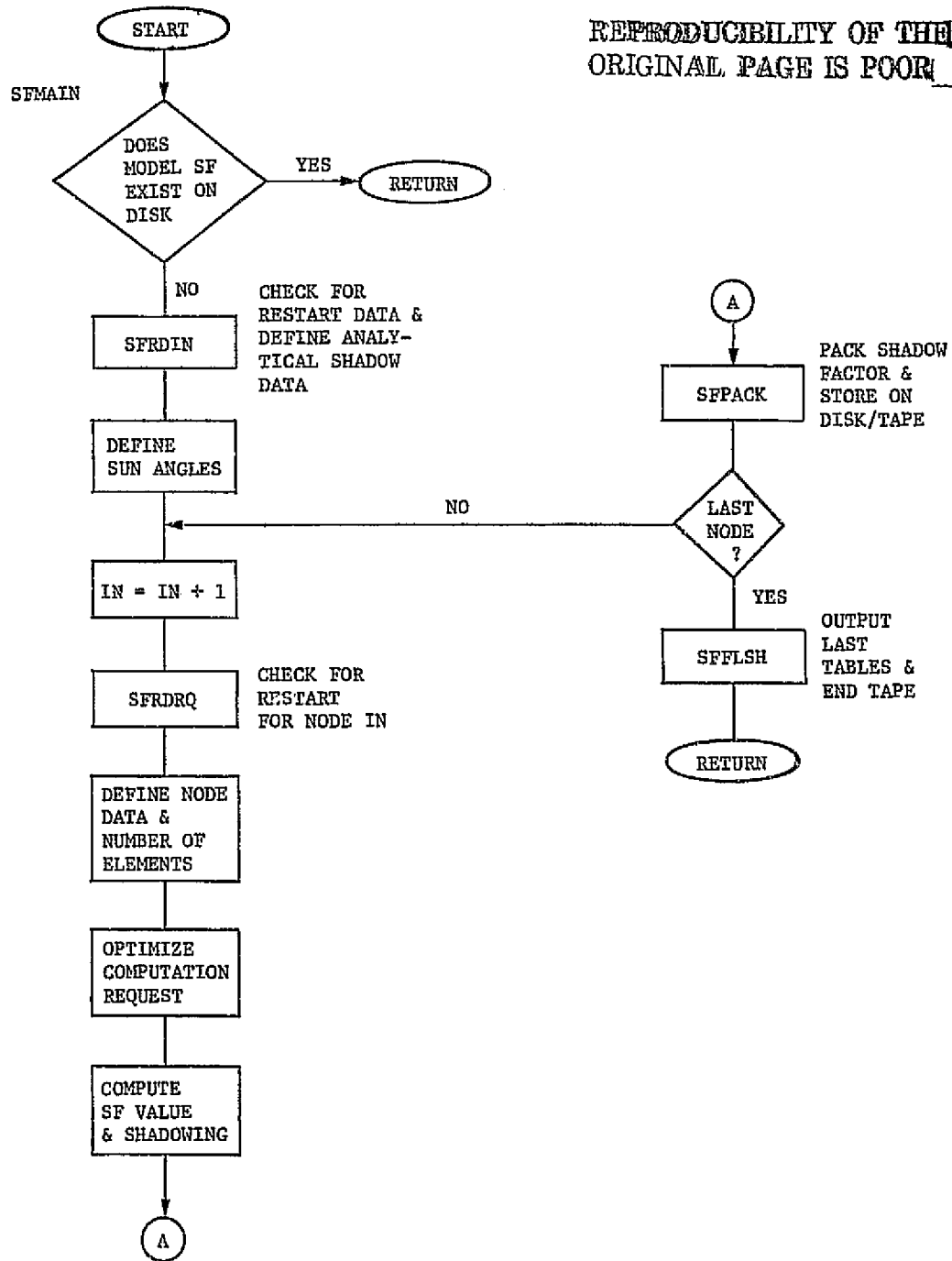


Figure III-13 Segment SFCAL Flow Diagram

B. SUBROUTINE AND FUNCTION DESCRIPTIONS - PROCESSOR LIBRARY

ROUTINE NAME: ABSBEA

DESCRIPTION: This routine locates plotting beam at the absolute raster coordinate (X, Y) based on 0.0 X, Y 1.0.

CALLING SEQUENCE: CALL ABSBEA (X, Y)

REFERENCES: NPMAIN, NPINFO, OPMAIN, OPINFO, PLGRID

ROUTINE NAME: ADSURF

DESCRIPTION: This routine provides the user with the capability to add an adiabatic "closure" surface to the problem geometry.

CALLING SEQUENCE: CALL ADSURF (NBCSN, NCONFIG, ARAS)

NBCSN - Name of block coordinate system under which adiabatic surface is defined

NCONFIG - Configuration name to locate appropriate form factor matrix

ARAS - Area of adiabatic surface

REFERENCE: ODPROG (Users operations data)

ROUTINE NAME: ADD

DESCRIPTION: This is an entry point in subroutine BUILDG that adds additional nodes/surfaces to the existing configuration (see BUILDG).

CALLING SEQUENCE: CALL ADD (BCSN)

BCSN - Left-justified, blank-filled block coordinate system name

REFERENCES: User call in the Operations Data Block

FILES: NRAN

ROUTINE NAME: AQCMPR

DESCRIPTION: This routine combines absorbed solar, albedo and planetary heat rate data according to correspondence data.

CALLING SEQUENCE: CALL AQCMPR

REFERENCE: AQMAIN

FILES: NSCR1, NGBIRR, NOUT

ROUTINE NAME: AQDATA

DESCRIPTION: This is a user-called routine to define control constants.

CALLING SEQUENCE: CALL AQDATA (IGBI, IGBS, RS, RA, RP)

IGBI - Step number containing gray-body IR  
IGBS - Step number containing gray-body SOLAR  
RS - Solar multiplier  
RA - Albedo multiplier  
RP - Planetary multiplier

REFERENCES: ODPROG (user's operations data)

ROUTINE NAME: AQEND

DESCRIPTION: Routine for user logic before termination of AQCAL link.

REFERENCES: AQMAIN

ROUTINE NAME: AQHEAD

DESCRIPTION: This routine writes to output file the default parameters and flags set for AQCAL execution.

CALLING SEQUENCE: CALL AQHEAD

REFERENCE: AQMAIN

ROUTINE NAME: AQMAIN

DESCRIPTION: This routine computes absorbed Q data from direct incident data and solar/IR gray bodies and stores the data on file NTQ.

CALLING SEQUENCE: CALL AQMAIN

KEY VARIABLES: QAS - Absorbed solar array  
QAR - Absorbed albedo array  
QAP - Absorbed planetary array  
GBIR - Gray-body IR array  
GBSO - Gray-body solar array  
QDS - Direct solar  
QDR - Direct albedo  
QDP - Direct planetary

REFERENCES: AQPROG

FILES: Reads NGBIR and NGBSO, NDI writes NTQ  
NOUT, NRSO

ROUTINE NAME: AQNCHK

DESCRIPTION: This routine checks the node array on the data file with the active node array to determine if they contain the same configuration. If the configurations are different, ACNCHK outputs both arrays and aborts, if both configurations are the same, it returns to the main program.

CALLING SEQUENCE: CALL AQNCHK (ITEMP, N1)

ITEMP - Array of node numbers from data file  
N1 - Number of nodes

KEY VARIABLES: NODE - Node number array of actual configuration  
NNOD - Number of active nodes in configuration

REFERENCES: AQMAIN

ROUTINE NAME: AQPRE

DESCRIPTION: This is a user-called routine in the AQCAL link prior to any computations.

CALLING SEQUENCE: CALL AQPRE

REFERENCES: AQMAIN

ROUTINE NAME: AQPRT

DESCRIPTION: This routine prints out the solar, albedo and planetary and total absorbed heat rates.

CALLING SEQUENCE: CALL AQPRT (IN)

IN - Node sequence number

REFERENCE: AQCMR

FILE: NOUT

ROUTINE NAME: AQPROG

DESCRIPTION: This routine is generated by the preprocessor to call into AQMAIN and the remainder of the link.

CALLING SEQUENCE: CALL AQPROG

KEY VARIABLES:   AQQDS   - Long direct solar in array NNOD  
                  AQQDR   - Long direct albedo in array NNOD  
                  AQQDP   - Long direct planetary in array NNOD  
                  QAS      - Long absorbed solar in array NNOD  
                  QAR      - Long absorbed albedo in array NNOD  
                  QAR      - Long absorbed planetary in array NNOD  
                  GBSO     - Long gray-body solar in array NNOD  
                  GBIR     - Long gray-body IR in array NNOD  
                  AQTEMP   - Working storage area in array NNOD

REFERENCES: TRASYS (root segment)

ROUTINE NAME: BANNLB

DESCRIPTION: This routine writes the TRASYS processor banner on the output file.

CALLING SEQUENCE: CALL BANNLB

REFERENCES: RDMAIN



ROUTINE NAME: BLDPLT

DESCRIPTION: This routine reads plot data from scratch disk 1 (NSCR1) and writes plot data in the proper format on output and plot disk IPLUNT. It also scales the X and Y values.

CALLING SEQUENCE: CALL BLDPLT (DATA, NV, NTIME, TIME)

DATA	- Working storage array
NV	- Number of lines per grid
NTIME	- Number of data points per line
TIME.(1, x)	- Independent variable (time)
TIME (2, x)	- Associated step number

REFERENCES: PLLOAD

FILES: NSCR1 is defined in PLLOAD. IPLUNT is written in plot format.  
NOUT, NSCR2

ROUTINE NAME: BUILDC

DESCRIPTION: This routine activates and initializes the configuration 1 block coordinate system name per call. It also defines arrays NODE, AREA, EMISS, TRIR, TRSO, SRIR, and SRSO, and initializes configuration counters NN, NS, NSPEC, and NONLY.

CALLING SEQUENCE: CALL BUILD (BCSN)/CALL ADD (BCSN)

BCSN - Left-justified, blank-filled block coordinate system name

KEY VARIABLES: INDXN - Array of random access record pointers to active nodes defined by BUILDG/ADD  
INDXS - Array of random access record pointers to active surfaces defined by BUILDG/ADD  
NN - Number of nodes plus shadower-only nodes  
NS - Number of surfaces  
NSPEC - Number of specular surfaces  
NNOD - Number of nodes  
NSURF - Number of surfaces  
NONLY - Number of shadower only surfaces

REFERENCES: User-called in the operations data block only

FILES: NRAN, NOUT

ROUTINE NAME: CHGBLK

DESCRIPTION: This routine reads block coordinate system data (translations and rotations) from random access file NRAN, redefines them, rewrites them to the NRAN file, and flags the BCS directory that the BCS has been changed.

CALLING SEQUENCE: CALL CHGBLK (NAME, X, Y, Z, NROTX, NROTY, NROTZ, ROTX, ROTY, ROTZ)

NAME	- BCS name	NROTX	- Rotation order of ROTX
X	- Translation X	NROTY	- Rotation order of ROTY
Y	- Translation Y	NROTZ	- Rotation order of ROTZ
Z	- Translation Z	ROTX	- Rotation angle about X-axis
		ROTY	- Rotation angle about Y-axis
		ROTZ	- Rotation angle about Z-axis

KEY VARIABLES: NBLKDR (1, I) - BCS name  
(2, I) - Length of block  
(3, I) - Random record number  
(4, I) - Flag transform has been applied

REFERENCES: User-called in the Operations Data block

FILES: NRAN - Rewritten  
NOUT

ROUTINE NAME: CLEAR

DESCRIPTION: This routine frees, clears and returns the data written on a unit so it may be reused. (Generally used with scratch files.)

CALLING SEQUENCE: CALL CLEAR (IUNIT)

IUNIT = Unit number to be freed

REFERENCED BY: All processor segments

ROUTINE NAME: CMCMBN

DESCRIPTION: This routine combines form factors, area and property arrays.

CALLING SEQUENCE: CALL CMCMBN (NODEC,AREAC,EMISSC,ALPHC,TRIRC,TRSDC,SRIRC,SRSOC,NLEN)

NODEC - Combined node array  
AREAC - Combined area array  
EMISSC - Combined emissivity array  
ALPHC - Combined solar absorptivity array  
TRIRC - Combined infrared transmissivity array  
TRSOC - Combined solar transmissivity array  
SRIRC - Combined infrared specular reflectivity array  
SRSOC - Combined solar specular reflectivity array  
NLEN - Maximum length of form factor record on NFF

REFERENCE: CMMAIN

FILES: NSCR1, NSCR2, NSCR3, NFF, NOUT, NRSI, NRSO

ROUTINE NAME: CMDATA

DESCRIPTION: This routine is called by the user to set up control constants for the CMCAL program segment.

CALLING SEQUENCE: CALL CMDATA (NFIG,NFCO,NTYP,IAUTO,FFCM,FFPRT)

KEY VARIABLES: Reference Users Manual, Appendix D

REFERENCE: ODPROG (User's Operations Data)

ROUTINE NAME: CMESUM

DESCRIPTION: This routine writes combined form factor sums to the output file.

CALLING SEQUENCE: CALL CMESUM (NNOD, NODE)

NNOD - Number of nodes after combining  
NODE - Combined node array

REFERENCE: CMCMBN, CMRSI

FILES: NOUT

ROUTINE NAME: CMINIT

DESCRIPTION: This routine sets up temporary node, area and property arrays and checks for duplicate node numbers.

CALLING SEQUENCE: CALL CMINIT (NODEC,EMMISSC,ALPHC,TRIRC,TRSOC,  
SRIRC,SRSOC)

KEY VARIABLES: Reference subroutine CMCMBN

REFERENCE: CMMAIN

FILES: NGBIRR, NOUT, NSCR3, NRSI

ROUTINE NAME: CMMAIN

DESCRIPTION: Driver routine for form factor combining segment.

CALLING SEQUENCE: CALL CMMAIN

REFERENCE: CMPROG

FILES: NOUT, NFF, NRSI

ROUTINE NAME: CMOUT

DESCRIPTION: This routine writes individual combined form factors to the output file.

CALLING SEQUENCE: CALL CMOUT (IN, JN, NODE, AREA, NNCM)

IN - Row number  
JN - Column number  
NODE - Node array  
AREA - Area array  
NNCM - Number of nodes after combining

REFERENCE: CMCMBN, CMRSI

FILE: NOUT

ROUTINE NAME: CMFPROG

DESCRIPTION: This is the preprocessor-generated routine that calls  
in the form factor combining segment.

REFERENCE: TRASYS ( ot segment)

ROUTINE NAME: CMRSI

DESCRIPTION: This routine reads combined form factors from the  
master restart tape.

CALLING SEQUENCE: CALL CMRSI (ICOMP, NODEC, AREAC, EMISSC, ALPHC,  
TRIRC, TRSOC, SRIRC, SRSOC)

KEY VARIABLES: Combined node, area and property arrays (reference  
subroutine CMCMBN)

REFERENCE: GMMAIN

FILES: NRSI, NOUT, NFF, NRSO, NSCR1

ROUTINE NAME: CMRSUM

DESCRIPTION: This routine writes combined form factor sums to  
the output file.

CALLING SEQUENCE: (NODE, AREA, SUM, IN)

NODE - Combined node array  
AREA - Combined area array  
SUM - Array of combined form factor sums  
IN - Node sequence numbers

REFERENCE: CMCMBN, CMRSI

FILE: NOUT

ROUTINE NAME: CMSAVE

DESCRIPTION: This routine writes the combined node, area and property arrays to a random access file.

CALLING SEQUENCE: CALL CMSAVE (ISAVE)

ISAVE - Random file record length flag

REFERENCE: CMCMBN, CMRSI

FILES: NRARR, NOUT

ROUTINE NAME: CMTRAN

DESCRIPTION: This routine transposes the combined form factor matrix.

CALLING SEQUENCE: CALL CMTRAN (DATA(1), DATA(N1), DATA(1), DATA(N1),  
DATA(N2), DATA(N3), DATA(N2), DATA(N3), NB, NND,  
NR, NRECT)

DATA - Array containing form factors  
NB - Number of blocks  
NND - Number of nodes  
NR - Number of records per block  
NRECT - Number of records, total

REFERENCE: CMMAIN

FILES: NFF, NSCR1, NSCR2

ROUTINE NAME: CROSS

DESCRIPTION: This routine forms a vector cross-product of the form.

$$\text{CROS} = \text{VEC1} \times \text{VEC2}$$

CALLING SEQUENCE: CALL CROSS (VEC1, VEC2, CROS)

VEC1 (3) - X, Y, Z in array

VEC2 (3) - X, Y, Z in array

CROS (3) - X, Y, Z in array

REFERENCES: DILOC2

ROUTINE NAME: DICALP

DESCRIPTION: This routine computes the direct incident albedo radiation QDR (IN) and direct planetary radiation QDP (IN), forms an optimum elemental grid based on altitude and node positions/shadowing, and saves elemental planetary factors on disk and applies shadowing. It is the main control routine for computing albedo and planetary radiation.

CALLING SEQUENCE: CALL DICALP (IRSPH)

KEY VARIABLES: SHADR - Shadow factor for albedo  
SHADP - Shadow factor for planetary  
SUMR - Elemental albedo factors  
SUMP - Elemental planetary factors  
QDR (IN) - Complete incident albedo for Node IN  
QDP (IN) - Complete incident planetary for Node IN  
ISFT - Flag to indicate shadow factor tape  
PLTYPE - Flag to indicate to save or read planet  
elemental factors

REFERENCES: DITYPE

FILES: NPLS, NOUT, NRSO

ROUTINE NAME: DICALS

DESCRIPTION: This is the main controlling routine to compute direct solar incident fluxes. It forms an optimum elemental grid based on position and shadowing.

CALLING SEQUENCE: CALL DICALS

KEY VARIABLES: ISFT - Flag to indicate shadow factor tape  
SHADS - Shadow factor  
QDS (IN) - Incident solar flux for Node IN

REFERENCES: DIMAIN

FILES: None

ROUTINE NAME: DICCV

DESCRIPTION: This routine defines the position vector, given clock and cone angles and the vector magnitude.

CALLING SEQUENCE: CALL DICCV (VMAG, CL, CO, PVEC)

VMAG - Magnitude  
CL - Clock angle in degrees  
CO - Cone angle in degrees  
PVEC(3) - Position vector

REFERENCES: DILOC2

ROUTINE NAME: DICLCO

DESCRIPTION: This routine computes clock and cone angles given a position vector.

CALLING SEQUENCE: CALL DICLCO (X, Y, Z, CL, CO)

X, Y, Z - Components of position vector  
CL, CO - Clock and cone angles

REFERENCE: DICALP, DICALS, DILOC, DIPRNT



ROUTINE NAME: DICOMB

DESCRIPTION: This routine performs a matrix multiplication of the form

$$A \cdot B = C$$

where

A, B, and C are 3 x 3 matrices.

CALLING SEQUENCE: CALL DICOMB (A, B, C)

A - 3 x 3 matrix

B - 3 x 3 matrix

C - Resultant 3 x 3 matrix

REFERENCES: DILOC, DILOC2

ROUTINE NAME: DICOMP

DESCRIPTION: This is a user routine to define compute flag or step numbers.

CALLING SEQUENCE: CALL DICOMP (IS, IA, IP)

IS - Solar flag to compute, zero, and retrieve data

IA - Albedo flag to compute, zero, and retrieve data

IP - Planetary flag to compute, zero, and retrieve data

REFERENCES: User-called routines in the Operations Data block.

ROUTINE NAME: DIDCS

DESCRIPTION: This routine computes direction cosines, given the Euler angles and the order to perform the rotations.

CALLING SEQUENCE: CALL DIDCS (II, JJ, KK, PHI, PSI, OMI, TRAN)

II	}	- Order of rotations
JJ		for angles PHI, PSI and OMI, respectively
KK		
PHI		- Rotation angle about Z-axis
PSI		- Rotation angle about Y-axis
OMI		- Rotation angle about X-axis
TRAN (3,3)		- Resultant direction-cosine matrix

REFERENCES: None

ROUTINE NAME: DIDENT

DESCRIPTION: This routine generates a 3 x 3 identity matrix in A.

CALLING SEQUENCE: CALL DIDENT (A)

A - 3 x 3 array with  $i = j = 1$ ,  $i \neq j = 0$

REFERENCES: DILOC, DIDT2, SURFP

ROUTINE NAME: DIDT1

DESCRIPTION: This is a user-called routine to define the DI link parameters.

CALLING SEQUENCE: See users manual, Appendix D

REFERENCES: User-called routine in the Operations Data block

ROUTINE NAME: DIDT1S

DESCRIPTION: This is a user-called routine to define the DI link parameters.

CALLING SEQUENCE: See users manual, Appendix D

REFERENCES: User called routine in the Operations Data block

ROUTINE NAME: DIDT2

DESCRIPTION: This is a user-called routine to define the DI link parameters.

CALLING SEQUENCE: See users manual, Appendix D

REFERENCES: Operations Data, user-called

ROUTINE NAME: DIDT2S

DESCRIPTION: Short form call to define DI link parameters.

CALLING SEQUENCE: See users manual, Appendix D

REFERENCES: User-called routine in the Operations Data block

ROUTINE NAMES: DIDT3, DIDT3S

DESCRIPTION. User called routines to define DICAL segment parameters.

CALLING SEQUENCES: See Users Manual, Appnedix D.

REFERENCES: ODPROG (User's Operations Data)

ROUTINE NAME: DIELEM

DESCRIPTION: Computes position and area vectors for all elements on a surface given the number of elements in each direction on a node.

CALLING SEQUENCE: CALL DIELEM (ILP, DATA, TRAN, RX, RY, RZ, NTOT)

ILP            - Type of geometric node  
DATA(5)       - Dimensions of node  
TRAN(3,3)     - Direction cosines of node  
RX, RY, RZ    - Translation vector  
NTOT          - Number of elements node is to be divided  
              into

KEY VARIABLES: NEST            - Number of elements (maximum = 400)  
              SFPV (400, 3) - Node elemental position vector  
              SFAV (400, 3) - Node elemental normal vector

REFERENCES: DICALS, DICALP

ROUTINE NAME: DIELSL

DESCRIPTION: This routine computes the number of elements in two directions on a node in an attempt to provide square elements, given the total number required.

CALLING SEQUENCE: CALL DIELSL (NB, NG, ILP, DATA, NTOT)

NB            - Computed number of elements in beta direction  
NG            - Computed number of elements in gamma direction  
ILP           - Surface type  
DATA(5)       - Nodal dimensions  
NTOT          - Total number of elements required

REFERENCES: DIELEM

ROUTINE NAME: DIEND

DESCRIPTION: This routine outputs final arrays of data (QDS - incident solar, the QDR - incident albedo, and QDP - incident planetary) to the NDI disk file for later reference. It end-files NRTO if a restart tape is desired and end-files NPLS if a PLTYPE (save) flag has been set.

CALLING SEQUENCE: CALL DIEND

REFERENCES: DIMAIN

FILES: Writes the NDI file and closes NPLS

REV 1  
III-34

ROUTINE NAME: DIENDP

DESCRIPTION: This is a user-callable routine to intervene after completing albedo/planetary computations.

CALLING SEQUENCE: CALL DIENDP

REFERENCES: DIMAIN

ROUTINE NAME: DIENDS

DESCRIPTION: This is a user-callable routine to intervene after completing solar computations.

CALLING SEQUENCE: CALL DIENDS

REFERENCES: DIMAIN

ROUTINE NAME: DIGTST

DESCRIPTION: This integer function routine determines if  $GN < GT < GX$ . If true, the function is set equal to 0; if false the value is set equal to 1.

CALLING SEQUENCE: I = DIGTST (GN, GX, GT)

GN - Lower bound  
GX - Upper bound  
GT - Intermediate value  
I - 0, 1 flag

REFERENCES: DISHAD

ROUTINE NAME: DILOC

DESCRIPTION: This routine computes necessary orbital parameters from input values for PERIOD, TRUEAN, TIMEPR, SUNPV, and various orbital transformations.

CALLING SEQUENCE: CALL DILOC

KEY VARIABLES: PERIOD - Orbit period  
TRUEAN - True anomaly  
TIMEPR - Present time  
SUNPV - Sun position vector  
PLDC - Matrix of direction cosines to transform vectors  
in the planet-oriented VCS to the user-defined VCS

REFERENCES: DIMAIN

FILES: NOUT - System output file

ROUTINE NAME: DILOC2

DESCRIPTION: This routine computes necessary orbital parameters, given the clock and cone angles to the sun and planet.

CALLING SEQUENCE: CALL DILOC2

KEY VARIABLES: SUNC<sub>L</sub> - Sun clock angle  
SUNCO - Sun cone angle  
PLCL - Planet clock angle  
PLCO - Planet cone angle  
IORBIT - Flag for type of orbit

REFERENCES: DILOC

FILES: NOUT - System output file  
NSCR3 - Scratch file

ROUTINE NAME: DIMAIN

DESCRIPTION: This routine is the main driving logic for computing direct incident fluxes and includes the main computation loops.

CALLING SEQUENCE: CALL DIMAIN

KEY VARIABLES: IN - Sequence to node number currently being computed

REFERENCES: DIPROG (preprocessor-generated)

ROUTINE NAME: DIOUTP

DESCRIPTION: This routine allows the user to change the type or form of printed or punched data. It normally calls:

DIPRTP - To print the albedo/planetary fluxes  
DIPNHP - To punch the albedo/planetary fluxes  
DITPP - To write the albedo/planetary fluxes to the RTO  
tape in restart format

CALLING SEQUENCE: CALL DIOUTP

REFERENCES: DIMAIN

ROUTINE NAME: DIOUTS

DESCRIPTION: This routine allows the user to change the type or form of printed/punched data. It normally calls:

DIPRTS - To print the solar fluxes  
DIPNHS - To punch the solar fluxes  
DITPS - To write the solar fluxes to the RTO tape in the  
restart format

CALLING SEQUENCE: CALL DIOUTS

REFERENCES: DIMAIN

ROUTINE NAME: DIPLNS

DESCRIPTION: This routine computes planet position and area vectors based on the orbit and accuracy parameters, and determines the emissive power of the planet element and view factor from the element to the sun.

CALLING SEQUENCE: CALL DIPLNS (NPEL)

NPEL - Total number of desired elements on planet

KEY VARIABLES: PLPVT (400, 3) -- Array of planet position vectors  
PLAVT (400, 3) -- Array of planet-normal vectors (magnitude  
= area)  
ALBF (400) -- Array of planet-to-sun view factors  
PLNF (400) -- Array of planet-element emissive powers

REFERENCES: DICALP

ROUTINE NAME: DIPNHP

DESCRIPTION: This routine punches albedo and planetary fluxes in a format acceptable for restart if the DIPNCH flag has been set to 3HPUN.

CALLING SEQUENCE: CALL DIPNHP

KEY VARIABLES: DIPNHP - Flag to determine if punched cards are requested  
NODE -- Array of node numbers  
QDR -- Array of albedo values  
QDP -- Array of planetary values

REFERENCES: DIOUTP, DIPTPR

FILES: NPUN

ROUTINE NAME: DIPNHS

DESCRIPTION: This routine punches solar fluxes in a restart format complete with proper header cards generated on the initial call to the routine.

CALLING SEQUENCE: CALL DIPNHS

KEY VARIABLES: IST -- Flag to determine if this is the initial call

REFERENCES: DIOUTS, DIPTSR

FILES: NPUN



ROUTINE NAME: DIPREP

DESCRIPTION: This routine can be replaced by the user prior to computing albedo/planetary fluxes.

CALLING SEQUENCE: CALL DIPREP

REFERENCES: DIMAIN

ROUTINE NAME: DIPRES

DESCRIPTION: This routine can be replaced by the user prior to computing solar fluxes.

CALLING SEQUENCE: CALL DIPRES

REFERENCES: DIMAIN

ROUTINE NAME: DIPRNT

DESCRIPTION: This routine writes to output default and actual values for all DICAL segment control constants.

CALLING SEQUENCE: CALL DIPRNT

REFERENCE: DIMAIN

FILE: NOUT

ROUTINE NAME: DIPROG

DESCRIPTION: This is a preprocessor-generated routine that calls in main driving logic to perform the direct-irradiation computations.

CALLING SEQUENCE: CALL DIPROG

REFERENCES: TRASYS (Root Segment)

ROUTINE NAME: DIPRTP

DESCRIPTION: This routine prints albedo/planetary fluxes after each nodal computation. It may be overridden by the user if he desired to change the format.

CALLING SEQUENCE: CALL DIPRTP

KEY VARIABLES: INSHAD - Flag to determine if node is in the planet's shadow  
ICRD - Restart card number  
IN - Current node sequence number

REFERENCES: DIOUTP

ROUTINE NAME: DIPRTS

DESCRIPTION: This routine prints solar fluxes after each nodal computation. It may be overridden by the user if he desires to change the output format.

CALLING SEQUENCE: CALL DIPRTS

KEY VARIABLES: INSHAD - Flag to determine if node is in the planet's shadow  
ICRD - Restart card number  
IN - Current node sequence number

REFERENCES: DIOUTS

ROUTINE NAME: DIPSHP

DESCRIPTION: This routine determines possible shadowing surfaces between Node IN and the planet element.

CALLING SEQUENCE: CALL DIPSHP (RADJ, RADL, POSJ, POSI, NST, JST, IN, JUMP)

RADJ - Radius of sphere enclosing surface node  
RADL - Radius of sphere enclosing planet element  
POSJ - Position vector of surface sphere  
POSI - Position vector of planet element  
NST - Number of shadowing surfaces  
JST - Number of possible shadowers  
IN - Node sequence number being computed  
JUMP - Flag for using cylinder or cone technique

KEY VARIABLES: ISHAD - Array of possible shadowers  
JST - Number of possible shadowers

REFERENCES: DICALP, SFPL

ROUTINE NAME: DIPSHS

DESCRIPTION: This routine determines possible shadowing surfaces between Node IN and the sun.

CALLING SEQUENCE: CALL DIPSHS (RADS, POS, SUNP, NSURF, NSHAD, IN)

RADS - Radius of sphere encompassing Node IN  
POS - Position vector of center of encompassing sphere  
SUNP - Sun position vector  
NSURF - Number of shadowing surfaces  
NSHAD - Number of possible shadowing surfaces  
IN - Sequence number of node being computed

KEY VARIABLES: ISHAD - Array of shadowing surfaces

REFERENCES: DIGALS

ROUTINE NAME: DIPTPR

DESCRIPTION: This routine writes planetary fluxes obtained from the restart tape to the output file.

CALLING SEQUENCE: CALL DIPTPR

REFERENCE: DIRDRQ, DIOUTP

FILES: NOUT

ROUTINE NAME: DIPTSR

DESCRIPTION: This routine writes solar fluxes obtained from the restart tape to the output file.

CALLING SEQUENCE: CALL DIPTSR

REFERENCE: DIRDRQ

FILES: NOUT

ROUTINE NAME: DIRCOS

DESCRIPTION: This routine computes the direction cosines, given the rotation order and angles.

CALLING SEQUENCE: CALL DIRCOS (II, JJ, KK, PHI, PSI, OMI, TRAN)

II                    order of rotation for PHI, PSI,  
JJ                    and OMI, respectively.  
KK  
PHI                  - Rotation about Z (Y to X = positive)  
PSI                  - Rotatinn about Y (X to Z = positive)  
OMI                  - Rotation about X (Y to Z = positive)  
TRAN(3,3) - Resultant direction cosine matrix

REFERENCES: ORIENT, BUILDC

ROUTINE NAME: DIRDRQ

DESCRIPTION: This routine defines the restart request arrays and determines the proper initial data based on the compute flags. . It also determines if shadowing is computed directly or by a table lookup. If computed, DIRDRQ defines arrays with all-shadowing surface data.

CALLING SEQUENCE: CALL DIRDRQ

KEY VARIABLES: ISOLFL - Flag for computing, storing, or zeroing the solar fluxes  
IALBFL - Flag for computing, storing, or zeroing the albedo fluxes  
IPLAFL - Flag for computing, storing, or zeroing the planetary fluxes

REFERENCES: DIMAIN

FILES: NDIR, NDI, NPLS, NOUT, NRAN, NRSI, NRSO, NRTI, NRTO

ROUTINE NAME: DIRPSP

DESCRIPTION: This routine computes the radius, area, and position vector for the planet.

CALLING SEQUENCE: CALL DIRPSP

KEY VARIABLES: RADP     - Radius of the planet  
AREAPL     - Area of visible portion of planet  
POSP(3)     - Position vector of planet

REFERENCES: DIMAIN, DIPRNT

ROUTINE NAME: DIRTI

DESCRIPTION: This routine reads flux data from the temporary  
restart tape.

CALLING SEQUENCE: CALL DIRTI

REFERENCE: DIRDRQ

FILES: NOUT, NDI, NRTI

ROUTINE NAME: DIRTOP

DESCRIPTION: This routine writes planetary fluxes to the temporary  
restart file.

CALLING SEQUENCE: CALL DIRTOP

REFERENCE: DIOUTP

FILE: NRTO

ROUTINE NAME: DIRTOS

DESCRIPTION: This routine writes solar flux data to the temporary  
restart tape.

CALLING SEQUENCE: CALL DIRTOS

REFERENCE: DIOUTS

FILE: NRTO

ROUTINE NAME: DIRTP

DESCRIPTION: This routine reads planet-element factors from a  
previously stored DI call and defines the QDR and QDP  
arrays based on known planet factors.

CALLING SEQUENCE: CALL DIRTP

REFERENCES: DITYPE

FILES: NPLS

ROUTINE NAME: DISFTP

DESCRIPTION: This routine reads shadow factor tables from the NPLS file for use in computing direct fluxes.

CALLING SEQUENCE: CALL DISFTP(IST)

REFERENCES: DICALP, DICALS, DIMAIN

FILES: NPLS

ROUTINE NAME: DITIME

DESCRIPTION: This routine computes the true anomaly from time-dependent orbital characteristics.

CALLING SEQUENCE: CALL DITIME (TIME, PER, ECC, TRU)

TIME - Input orbital time  
PER - Orbital period  
ECC - Eccentricity of orbit  
TRU - Output true anomaly

REFERENCES: DILOC, STFAQ

ROUTINE NAME: DITRS3

DESCRIPTION: This routine converts points from the surface coordinate system to the central coordinate system.

CALLING SEQUENCE: CALL DITRS3 (X, Y, Z, A, B, C, RX, RY, RZ, TRAN)

X, Y, Z - Output components of vector in CCS  
A, B, C - Components of vector to be transformed  
RX, RY, RZ - Position vector of SCS origin in CCS  
TRAN - Direction cosine matrix

REFERENCES: DIELEM

ROUTINE NAME: DITRS4

DESCRIPTION: This routine converts points in the central coordinate system to the surface coordinate system.

CALLING SEQUENCE: CALL DITRS4 (X, Y, Z, A, B, C, RX, RY, RZ, TRAN)

X, Y, Z - Output components of vector in SCS  
A, B, C - Components of vector to be transformed  
RX, RY, RZ - Translation vector, in SCS  
TRAN - Direction cosine matrix

REFERENCES: DILOC, DILOC2

ROUTINE NAME: DITTP

DESCRIPTION: See program listing for definition.

ROUTINE NAME: DITYPE

DESCRIPTION: This routine determines the type of computation and calls the proper routine.

CALLING SEQUENCE: CALL DITYPE (IRSPH)

KEY VARIABLES: PLTYPE - Flag indicating that planetary data were previously defined.

REFERENCES: DIMAIN

ROUTINE NAME: DIVWPL

DESCRIPTION: This routine computes elemental planetary factors without shadowing, and forms a sum over all the elements.

CALLING SEQUENCE: CALL DIVWPL (NTOTP,AREASF,NOT,FRACT,FRACTR,FRACTP)

REFERENCES: DICALP

ROUTINE NAME: DIVWSN

DESCRIPTION: This routine computes unshadowed elemental view factors to the sun and forms a sum over all the elements.

CALLING SEQUENCE: CALL DIVWSN (AREASF,NOT,FRACT)

REFERENCES: DICALS

ROUTINE NAME: DUPNCK

DESCRIPTION: This is a user-called routine in the Operations Data block that determines if a configuration has duplicate node numbers. It aborts if duplicates are found.

CALLING SEQUENCE: CALL DUPNCK

REFERENCES: Operations Data block

ROUTINE NAME: DRCALS

DESCRIPTION: This is the main routine used in calculating the specularly reflected components of incident solar flux.

CALLING SEQUENCE: CALL DRCALS (MIRROR, KN, NSURFS)

MIRROR - Surface sequence number of current specular surface  
KN - Index ranging over the number of specular surfaces and indicating the current specular surface  
NSURFS - Total number of surfaces, plus images of surfaces, in MIRROR

REFERENCES: DRMAIN



ROUTINE NAME: DRDATA

DESCRIPTION: This routine is called by the user to set up necessary control constants to execute program segment.

CALLING SEQUENCE: CALL DRDATA (NSTP, DIACCLS)

KEY VARIABLES: Reference Users Manual, Appendix D.

Reference: ODPROG (User's Operations Data)

ROUTINE NAME: DRELEM

DESCRIPTION: Given the number of elements in each direction on a node, this routine calculates the area and position vectors for each element.

CALLING SEQUENCE: CALL DRELEM (ILP, DATA, TRAN, RX, RY, RZ, NTOT)

ILP	- Surface type
DATA	- Nodal dimension parameters (ALPHA, BMIN, BMAX, GMIN, and GMAX)
TRAN	- 3 x 3 matrix of direction cosines
RX, RY, RZ	- Translation vector
NTOT	- Total number of elements required on node

KEY VARIABLES: NEST - Counter for number of elements on a node  
(maximum allowable = 100)  
SFAV - Array of elemental surface area vectors  
SFPV - Array of elemental position vectors

REFERENCES: DRCALS

ROUTINE NAME: DRELSL

DESCRIPTION: Given the total number of elements on a node, this routine calculates the number of elements in each direction so as to make them as nearly square as possible.

CALLING SEQUENCE: CALL DRELSL (NB, NG, ILP, DATA, NTOT)

NB	- Number of elements in the beta direction
NG	- Number of elements in the gamma direction
ILP	- Surface type
DATA	- Nodal dimension parameters (ALPHA, BMIN, BMAX, GMIN, and GMAX)
NTOT	- Minimum number of elements to be distributed over node

REFERENCES: DRELEM

ROUTINE NAME: DREND

DESCRIPTION: This routine writes direct incident fluxes QDS, QDR, and QDP to the NDI disk file (labeled 6HIMAGEQ) to indicate that specular components are included.

CALLING SEQUENCE: CALL DREND

KEY VARIABLES: QDS - Incident solar flux  
QDR - Incident albedo flux  
QDP - Incident planetary flux

REFERENCES: DRMAIN

FILES: NDI - Disk file for storing direct incident fluxes  
NRSO

ROUTINE NAME: DRENDS

DESCRIPTION: This is a user routine that enables the user to intervene after calculating the incident solar fluxes.

CALLING SEQUENCE: CALL DRENDS

REFERENCES: DRMAIN

ROUTINE NAME: DRGTST

DESCRIPTION: This is an integer function routine that tests a given value, GT, to determine if it falls in the range  $GN < GT < GX$ . If true, the function value is 0 (zero); if false, the function value is 1 (one).

CALLING SEQUENCE: DRGTST (GN, GX, GT)

GN - Lower bound  
GX - Upper bound  
GT - Value to be tested

REFERENCES: DRSHAD

ROUTINE NAME: DRIMAG

DESCRIPTION: This routine images the solar vector, as well as all shadowing surfaces, and writes the results on scratch file NSCR2 for each specular surface.

CALLING SEQUENCE: CALL DRIMAG (NST)

NST	- Number of active surfaces
NSURFS	- Number of active surfaces plus number of images in any given specular surface
IFS, IKS, PR, DSTR, DIMS, PSH, TSTR	- Shadowing surface description parameters
SUNPVT	- Solar vector

REFERENCES: DRMAIN

FILES: NSCR2 - Scratch file

ROUTINE NAME: DRMAIN

DESCRIPTION: This routine contains the main driving logic for computing direct incident fluxes, including specular components.

CALLING SEQUENCE: CALL DRMAIN

KEY VARIABLES: ISPEC - Array of surface sequence numbers for specular surfaces  
SUNPVT - Solar vector  
PLDC - Matrix of direction cosines to transform vectors in the planet-oriented VCS to the user-defined VCS

REFERENCES: DRPROG

FILES: NRAN - Random access file  
NSCR2 - Scratch file  
NDI, NOUT, NRSI, NRTO

ROUTINE NAME: DROUTP

DESCRIPTION: This is a user routine that can be replaced to change the form of output for albedo and planetary incident fluxes. The normal call is to DRP RTP, which prints the albedo and planetary fluxes.

CALLING SEQUENCE: CALL DROUTP

REFERENCES: DRMAIN

ROUTINE NAME: DROUTS

DESCRIPTION: This is a user routine that can be replaced to change the form of output for incident solar fluxes. The normal call is to DRP RTS, which prints the solar fluxes.

CALLING SEQUENCE: CALL DROUTS

REFERENCES: DRMAIN

ROUTINE NAME: DRPOSI

DESCRIPTION: This routine transforms a vector in the ICS, BCS, or CCS to a vector in the SCS of a specular surface, negates the Z component, and transforms it back into the ISC, BCS, or CCS.

CALLING SEQUENCE: CALL DRPOSI (X, Y, Z, TRAN)

X, Y, Z - Vector components in ICS, BCS, or CCS  
TRAN - Matrix of direction cosines

KEY VARIABLES: A, B, C - Vector components in the SCS

REFERENCES: DRIMAG

ROUTINE NAME: DRPRT

DESCRIPTION: This routine prints albedo and planetary incident fluxes after each nodal computation. The user can override this routine to change the output format if he desires.

CALLING SEQUENCE: CALL DRPRT

KEY VARIABLES: IN - Current node sequence number  
INSHAD - Flag to indicate if vehicle is in planet shadow

REFERENCES: DROUTP

FILES: NOUT - System output file

ROUTINE NAME: DRPRTS

DESCRIPTION: This routine prints the solar incident flux after each nodal computation. The user can override this routine to change the output format if he desires.

CALLING SEQUENCE: CALL DRPRTS

KEY VARIABLES: IN - Current node sequence number  
INSHAD - Flag to indicate if vehicle is in planet shadow

REFERENCES: DROUTS

FILES: NOUT - System output file

ROUTINE NAME: DRPSHS

DESCRIPTION: This routine determines all possible shadowing surfaces between Node IN and the image of the sun in specular surface MIRROR.

CALLING SEQUENCE: CALL DRPSHS (RADS, POS, SUNP, NSURF, NSS, NSHAD, IN, MIRROR)

RADS - Radius of sphere enclosing Node IN  
POS - Position vector of enclosing sphere  
SUNP - Image of solar vector in specular surface MIRROR  
NSURF - Total number of shadowing surfaces  
NSS - Total number of shadowing surfaces plus images of shadowing surfaces in MIRROR  
NSHAD - Number of possible shadowing surfaces  
IN - Sequence number of current node  
MIRROR - Surface sequence number of current specular surface

REFERENCES: DRCALS

ROUTINE NAME: DRRDRO

DESCRIPTION: This routine initializes the direct incident flux arrays QDS, QDR, and QDP to the flux values calculated by segment DICAL.

CALLING SEQUENCE: CALL DRRDRO (KST)

KEY VARIABLES: QDS - Incident solar flux array  
QDR - Incident albedo flux array  
QDP - Incident planetary flux array

REFERENCES: DRMAIN

FILES: NOUT - System output file  
NDI - Disk file for storing direct incident fluxes

ROUTINE NAME: DRRSI

DESCRIPTION: This routine reads real-body effect direct fluxes from the restart tapes.

CALLING SEQUENCE: CALL DIRSI (ICOMP, KN)

ICOMP - Compute/no compute flag  
KN - Node number on RTI tape record

REFERENCE: DRMAIN

FILES: NRSI, NRTI

REV 1  
III-52

ROUTINE NAME: DRSHAD

DESCRIPTION: This routine calculates the elemental shadowing, where an element is either completely shadowed or not shadowed at all. Shadowing is reduced by the transmissivity for semitransparent shadowing surfaces.

CALLING SEQUENCE: CALL DRSHAD (ILKI, RX, RY, RZ, WE, WA, IN, I, NSS, RS, MIRROR)

RX, RY, RZ	- Components of solar position vector
WE, WA	- Elemental shadowing factors for IR and solar fluxes
IN	- Surface sequence number
I	- Sequence number of element
NSS	- Number of possible shadowing surfaces
RS	- Square of the magnitude of the solar vector
MIRROR	- Sequence number of current specular surface

REFERENCES: DRCALS

ROUTINE NAME: DRTRS3

DESCRIPTION: This routine transforms points in an SCS to points in the ICS, BCS, or CCS.

CALLING SEQUENCE: CALL DRTRS3 (X, Y, Z, A, B, C, RX, RY, RZ, TRAN)

X, Y, Z	- Coordinates of a point in the ICS, BCS or CCS
A, B, C	- Coordinates of the point in the SCS
RX, RY, RZ	- Components of the SCS origin position vector in the ICS, BCS, or CCS
TRAN	- Matrix of direction cosines

REFERENCES: DRELEM, DRIMAG, DRSHAD

ROUTINE NAME: DRTRS4

DESCRIPTION: This routine transforms points in the ICS, BCS, or CCS to points in an SCS.

CALLING SEQUENCE: CALL DRTRS4 (X, Y, Z, A, B, C, RX, RY, RZ, TRAN)

X, Y, Z - Coordinates of a point in the SCS  
A, B, C - Coordinates of the point in the ICS,  
          BCS, or CCS  
RX, RY, RZ - Components of the ICS, BCS, or CCS  
            origin position vector in the SCS  
TRAN - Matrix of direction cosines

REFERENCES: DRIMAG

ROUTINE NAME: DRVWSN

DESCRIPTION: This routine calculates the unshadowed form factor from a node to the sun.

CALLING SEQUENCE: CALL DRVWSN (AREASF, NOT, FRACT)

AREASF - Surface area of current node  
NOT - Flag to indicate whether node can "see"  
      the sun  
      = 0 Can see  
      = 1 Cannot see  
FRACT - Unshadowed node-to-sun form factor

REFERENCES: DRCALS

ROUTINE NAME: EARTH D

DESCRIPTION: This routine defines all Earth-oriented constants needed by the DI link.

CALLING SEQUENCE: CALL EARTH D

KEY VARIABLES: PRAD - Planet Radius  
                  SOL - Solar Constant  
                  PALB - Planet Albedo Factor  
                  WDS - Planet Darkside Emissive Power  
                  WSS - Planet Subsolar Emissive Power  
                  GRAV - Gravitational Constant



REFERENCES: ORBIT1, ORBIT2, DITTP

ROUTINE NAME: ENDTP

DESCRIPTION: This routine writes a pseudo end file on the BCD unit and informs the user as to type of data the unit contains.

CALLING SEQUENCE: CALL ENDTP (NTYPE)

NTYPE - Flag to indicate type of end message to write

REFERENCES: QOSAVE, QOCOMB, RCMAN

ROUTINE NAME: FFAREA

DESCRIPTION: This function routine computes the area of a surface, given the surface properties.

CALLING SEQUENCE: AC = FFAREA (ILP, ALPH, BMIN, BMAX, GMIN, GMAX)  
AC - Surface area  
ILP - Surface type  
ALPH, BMIN, BMAX, GMIN, GMAX - Surface dimensions

REFERENCES: FFEXPAN, FFEXPNI

ROUTINE NAME: FFCAL

DESCRIPTION: This is the main computation routine for computing the form factor. It determines the proper number of elements and computes position area vectors and the final form factor, including shadowing.

CALLING SEQUENCE: CALL FFCAL

REFERENCES: FFPRE

ROUTINE NAME: FFCALI

DESCRIPTION: This routine is analagous to subroutine FFCAL. It is used for form calculation when reversing the order of the i - j node pair due to area difference.

CALLING SEQUENCE: CALL FFCALI

REFERENCE: FFPRE

ROUTINE NAME: FFDATA

DESCRIPTION: This is a user-callable routine to define the control variables in the FF link.

CALLING SEQUENCE: CALL FFDATA (ACC, ACCS, IRNOSH, RATL, RMIN, IPRNT, IPUN, IFNAC)

See Users Manual, Appendix D, for definition

REFERENCES: Operations Data block

ROUTINE NAME: FFELEM

DESCRIPTION: This routine computes elemental position and area vectors, given the number required in each nodal direction.

CALLING SEQUENCE: CALL FFELEM (NB, NG, ILP, IC, DATA, POS, ARA, TRAN, RX, RY, RZ)

NB	- Number of elements in the beta direction
NG	- Number of elements in the gama direction
ILP	- Surface type
IC	- Position to start storing element in the POS and ARA arrays
DATA(5)	- Array of node dimensions
POS(500, 3)	- Array of element position vectors
ARA(500, 3)	- Array of normal vectors (magnitude & area)
TRAN(3, 3)	- Direction cosine matrix of node
RX, RY, RZ	- Position vector of SCS origin in the CCS

REFERENCES: FFCAL, FFEXPAN, FFCALI, FFXPNI

REV 1

III-56

ROUTINE NAME: FFELSL

DESCRIPTION: This routine uses the total number of elements on a node to compute the number of elements in various directions so that the elements are as nearly square as possible.

CALLING SEQUENCE: CALL FFELSL (NB, NG, ILP, DATA, NTOT)

NB - Computed number of elements in the beta direction  
NG - Computed number of elements in the gamma direction  
ILP - Surface type  
DATA(S) - Array of surface dimensions  
NTOT - Total number of elements to subdivide the node into

REFERENCES: FFCAL, FFEXPN, FFCALI, FFELSL, FFXPNI

ROUTINE NAME: FFEND

DESCRIPTION: This routine is called prior to the termination of the FF link to write the end-of-files to the internal disk units and restart files.

CALLING SEQUENCE: CALL FFEND

REFERENCES: FFMAIN

FILES: NFF,

ROUTINE NAME: FFEQIV

DESCRIPTION: This routine retrieves equivalent form factors from data files when equivalent form factor flags are found in the form factor request matrix.

CALLING SEQUENCE: CALL FFEQIV (FTEMP)

FTEMP - Array for use in reading random-access file

REFERENCES: FFOUT

FILES: NOUT, NSCR3

ROUTINE NAME: FFESUM

DESCRIPTION: This routine outputs form-factor sums for all nodes.

CALLING SEQUENCE: CALL FFESUM

KEY VARIABLES: SUM - Array of form-factor sums

REFERENCES: FFEND

ROUTINE NAME: FFEXPN

DESCRIPTION: This routine computes a node-pair form factor with shadowing, using the subnode technique. Logic is entered when the separation distance along the node varies by more than FFRATL.

CALLING SEQUENCE: CALL FFEEXPN (POSI, ARAI, FE, FA, RATI, RATJ, JFLAG, JST, NEI, NEJ, NSUR, IADDR)

POSI	- Position vector, storing elements on node i and j
ARAI	- Area vector, storing elements on node i and j
FE	- Resultant IR form factor
FA	- Resultant solar form factor
RATI	- Ratio of rate of change along node i
RATJ	- Ratio of rate of change along node j
JFLAG	- Flag indicating that maximum number of elements has been exceeded
JST	- Number of possible shadowers
NEI	- Average number of elements on node i
NEJ	- Average number of elements on node j
NSUR	- Average number of shadowing surfaces
IADDR	- Maximum time return sequence (not used)

REFERENCES: FFCAL

ROUTINE NAME: FFGTST

DESCRIPTION: This integer function routine determines if gamma falls in allowable ranges and sets the functional value to 1 if it does and to 0 if it doesn't.

CALLING SEQUENCE: FFGTST (GN, GX, GT)

GN - Minimum gamma  
GX - Maximum gamma  
GT - Gamma in question

REFERENCES: FFSHD

ROUTINE NAME: FFHEAD

DESCRIPTION: This routine prints FFcontrol values in summary form on an output file.

CALLING SEQUENCE: CALL FFHEAD

REFERENCES: FFMAIN

ROUTINE NAME: FFMAIN

DESCRIPTION: This is the main controlling logic in the FF link.  
It checks for restart data and directs the logic flow.

CALLING SEQUENCE: CALL FFMAIN

REFERENCES: FFPROG

ROUTINE NAME: FFMINR

DESCRIPTION: This routine determines if the computed form factor is less than the control value FFMIN. If less, it redefines the FFVALS and FFVALI array elements to zero.

CALLING SEQUENCE: CALL FFMINR

REFERENCES: FFOUT

ROUTINE NAME: FFNDP

DESCRIPTION: This routine provides the user with a node array punched in form factor and flux data block format.

CALLING SEQUENCE: CALL FFNDP

REFERENCE: ODPROG (Users Operation Data)

FILES: NOUT

ROUTINE NAME: FFOUT

DESCRIPTION: This is a user-replaceable routine calling for print (as a minimum) and punch options.

CALLING SEQUENCE: CALL FFOUT

REFERENCES: FFMMAIN

ROUTINE NAME: FFPCH

DESCRIPTION: This routine punches restart form factors if the FFPNCH option is used. The restart form factors are punched in a format acceptable to the Header Form Factor Data block.

CALLING SEQUENCE: CALL FFPCH

REFERENCES: FFOUT

ROUTINE NAME: FFPRE

DESCRIPTION: This is a user-definable routine to determine the type of computation. The program normally calls FFCAL to use a finite-element technique.

CALLING SEQUENCE: CALL FFPRE

REFERENCES: FFMMAIN

ROUTINE NAME: FFPROG

DESCRIPTION: This is a preprocessor-generated routine that defines necessary common blocks and calls into the main logic.

CALLING SEQUENCE: CALL FFPROG

REFERENCES: TRASYS (root segment)

ROUTINE NAME: FFPRT

DESCRIPTION: This routine prints nonzero computed and predefined form factors on an output file.

CALLING SEQUENCE: CALL FFPRT

REFERENCES: FFOUT

ROUTINE NAME: FFPSHD

DESCRIPTION: This routine computes the number of possible shadowing surfaces between node pairs.

CALLING SEQUENCE: CALL FFPSHD (RADJ, RADI, POSJ, POSI, NST, JST, IN, JN)

RADJ - Radius of sphere enclosing node j  
RADI - Radius of sphere enclosing node i  
POSI - Position vector to center of sphere enclosing node i  
POSJ - Position vector to center of sphere enclosing node j  
NST - Number of shadowing surfaces  
JST - Number of possible shadowers  
IN - Sequence number of node i  
JN - Sequence number of node j

REFERENCES: FFCAL, FFEXPN, FFCALI, FFXPNI

ROUTINE NAME: FFRDIN

DESCRIPTION: This routine determines if there is a form factor restart, and defines the shadowing data.

CALLING SEQUENCE: CALL FFRDIN (ISHDQ, IRSC)

ISHDQ - Flag indicating shadower-only surfaces  
1 - Shadower-only surfaces  
0 - Not shadower-only surfaces

REFERENCES: FFMAIN

ROUTINE NAME: FFRDRQ

DESCRIPTION: This routine reads the restart file of form factors and defines known values, or sets a flag to compute values for the current row.

CALLING SEQUENCE: CALL FFRDRQ

IEOFFR - End-of-file flag for the NFFR unit  
FFVALI - Array of values/flags to compute IR form factors  
FFVALS - Array of values/flags to compute solar form factors

REFERENCES: FFMAIN

FILES: NOUT, NRSI, NRTI

REV 1  
III-62



ROUTINE NAME: FFRIN

DESCRIPTION: This routine reads form factor information from the form factor data block via file NFFR.

CALLING SEQUENCE: CALL FFRIN

REFERENCE: FFRDIN, FFRDRQ

FILE: NOUT

ROUTINE NAME: FFROW

DESCRIPTION: This routine outputs to internal file NFF a row of form factors to be used in matrix form by other links.

CALLING SEQUENCE: CALL FFROW

REFERENCES: FFMAIN, FFRDIN

FILES: Writes NFF  
NRSO, NRTO, NSCR3

ROUTINE NAME: FFRPSN

DESCRIPTION: Given the nodal dimensions, this routine computes the radius of an encompassing sphere and the position vector to the center of the sphere.

CALLING SEQUENCE: CALL FFRPSN (RADN, POSN, ILK, DATA, BETA, GAMMA, DB, DG)

RADN - Computed radius of sphere enclosing node  
POSN(3) - Vector to center of sphere  
ILK - Surface type  
DATA(5) - Dimensions of node  
BETA - Length to center along beta direction  
GAMMA - Length to center along gamma direction  
DB - Absolute value of deviation from gamma  
DG - Absolute value of deviation from beta

REFERENCES: FFEXPN, FFXPN1

ROUTINE NAME: FFRSMR

DESCRIPTION: This routine writes the form factor sums to the output file when form factors are read from the master restart tape.

CALLING SEQUENCE: CALL FFRSMR

REFERENCE: FFRDIN

FILE: NOUT

ROUTINE NAME: FFRSUM

DESCRIPTION: This routine is called after completing each row, to output the FF sum and time.

CALLING SEQUENCE: CALL FFRSUM

REFERENCES: FFMAIN

ROUTINE NAME: FFRTR

DESCRIPTION: This routine reads form factor data from the temporary restart tape.

CALLING SEQUENCE: CALL FFRTR

REFERENCE: FFRDIN

FILE: NRTI, NOUT

ROUTINE NAME: FFSHD

DESCRIPTION: This routine computes elemental shadowing between node IN and node JN. An element is either completely shadowed or not shadowed at all. Shadowing is reduced by the transmissivity for semitransparent shadowing surfaces.

CALLING SEQUENCE: CALL FFSHD (ILKI, ILKJ, RX, RY, RZ, POSI, WE, WA, JST, IN, JN, I, NSS, RS)

ILKI - Type of surface for IN  
RX, RY, RZ - Components of vector from element on node i to element on node j  
POSI - Array of elemental position vectors  
WE, WA - Elemental shadowing factors for IR and solar form factors  
IN, JN - Surface sequence numbers  
I - Element sequence number  
NSS - Number of possible shadowers  
RS - Square of the magnitude of the element-to-element connecting vector ( $RX^2 + RY^2 + RZ^2$ )

REFERENCES: FFCAL, FFEXPN, FFCALI, FFXPNI, FFPCL

ROUTINE NAME: FFTMCK

DESCRIPTION: This routine calculates the time remaining in the run and compares it with the estimated time for the next calculation. If there is insufficient time remaining, the run is aborted.

CALLING SEQUENCE: CALL FFTMCK (IADDR, MAXLFT)

IADDR - Not used  
MAXLET - Estimated time required for next calculation  
M - Time remaining in run

REFERENCES: FFCAL, FFEXPN, FFCALI, FFXPNI

ROUTINE NAME: FFTRS3

DESCRIPTION: This routine transforms a point in an SCS to points in the ICS, BCS, and CCS.

CALLING SEQUENCE: CALL FFTRS3 (X, Y, Z, A, B, C, RX, RY, RZ, TTAN)

X, Y, Z - Coordinates of a point in the ICS,  
BCS, or CCS  
A, B, C - Coordinates of the point in the SCS  
RX, RY, RZ - Components of the SCS origin position  
vector in the ICS, BCS, or CCS  
TRAN - Matrix of direction cosines

REFERENCES: FFELEM, FFEXPN, FFUNIT, FFUNTI, FFXPNI

ROUTINE NAME: FFUNIT, FFUNTI

DESCRIPTION: These routines calculate unshadowed form factors between quadrilaterals (and triangles) using the Nusselt sphere method with a single integration over the I-node.

CALLING SEQUENCE: CALL FFUNIT (NTI, POSI, ARAI, FRACT, NTOTI, NTOTJ, NOT)

KEY VARIABLES: See FFVIEW

REFERENCES: FFCAL, FFCALI

ROUTINE NAME: FFVIEW

DESCRIPTION: This routine calculates the unshadowed form factor between node IN and node JN. It also calculates RATI and RATJ, which will later be compared with the user-input variable, FFRATL, to see if nodes should be expanded into subnodes for better accuracy.

CALLING SEQUENCE: CALL FFVIEW (NTI, NTJ, POSI, ARAI, NTOTI, NTOTJ, ARI, ARJ, NOT, FRACT, RATI, RATJ)

NTI	- Initial number of elements on node IN
NTJ	- Total initial number of elements on node IN and node JN
POSI	- Array of elemental position vector
ARAI	- Array of elemental area vectors
NTOTI, NTOTJ	- Number of elements required on node IN and node JN
ARI, ARJ	- Areas of node IN and node JN
NOT	- Flag to indicate whether node IN can see node JN = 0 Can see = 1 Cannot see
FRACT	- Unshadowed node-to-node form factor
RATI, RATJ	- Ratios of indicated maximum elements required in NTOTI and NTOTJ

REFERENCES: FFCAL, FFCALI

ROUTINE NAME: FFVWT

DESCRIPTION: This routine calculates unshadowed form factors between subnodes and determines the number of elements required based on a weighted-average criterion.

CALL FFVWT (NTI, NTJ, POSI, ARAI, NTOTI, NTOTJ, ARI, ARJ, NOT, FRACT)

NTI	- Number of elements on subnode of node IN
NTJ	- Total number of elements on subnode IN and subnode JN
POSI	- Array of elemental position vectors
ARAI	- Array of elemental area vectors
NTOTI, NTOTJ	- Number of elements required on subnodes of node IN and node JN
ARI, ARJ	- Areas of subnodes
NOT	- Flag to indicate whether subnode IN can "see" subnode JN = 0 Can see = 1 Cannot see
FRACT	- Unshadowed subnode-to-subnode form factor

REFERENCES: FFEXPN, FFXPNI

ROUTINE NAME: FFXPNI

DESCRIPTION: This routine is analogous to subroutine FFEXPN and is used when the order of the i - j pair is reversed due to area difference.

CALLING SEQUENCE: SEE FFEXPN

REFERENCE: FFCALI

ROUTINE NAME: FINDST

DESCRIPTION: Given the unit, step number, label 1, and label 2, this routine positions the file at the proper point or returns and end-of-file flag indicating it was unable to find the data.

CALLING SEQUENCE: CALL FINDST (NUNIT, ISTEP, LAB1, LAB2, IEOF)

NUNIT - Unit to be searched  
ISTEP - Program step number needed  
LAB1 - Subidentifier 1  
LAB2 - Subidentifier 2  
IEOF - Flag indicating if data were found  
= 2HNO, No end-of-file encountered (found data)  
= 3HYES End-of-file found (no data)

REFERENCES: FFRDIN, SFMAIN, SFRDIN, DIRDRQ, GBMAIN, AQMAIN, QOMAIN, QOCMBN, PLLOAD, FNDFLP, RCMAIN, ADSURF, AQCMFR, CMINIT, CMMAIN, DIMAIN, DIRT1, DRRDRQ, GBAPRX, PLCMBN, RBRDIN, RCINIT, STFAQ

FILES: All internal data files

ROUTINE NAME: FLIP

DESCRIPTION: Given a working storage area, this routine flips a matrix in row order on unit NS1 to a matrix in column order and writes it to unit NS2.

CALLING SEQUENCE: CALL FLIP (DATA, NV, NTIME, ICOL, NS1, NS2)

DATA - Working storage area  
NV - Number of rows stored at one time  
NTIME - Number of rows  
ICOL - Number of columns  
NS1 - Unit containing the input matrix  
NS2 - Unit containing the output matrix

REFERENCES: FNDFLP

ROUTINE NAME: FNDEXP

DESCRIPTION: Given a floating-point number, this routine converts it to scientific notation.

CALLING SEQUENCE: CALL FNDEXP (X, BASE, IEXP)

X  
BASE -  $0.0 \leq \text{BASE} \leq 1.0$   
IEXP - Exponent such that  $X = \text{BASE}^{**}\text{IEXP}$

REFERENCES: QOSBCD, QOAVGS

ROUTINE NAME: FNDFLP

DESCRIPTION: This routine reads the overall incident solar, albedo, planetary, and total fluxes from a disk generated by AQCAL or DICAL and stores them on a scratch file by sequential orbit points. It also builds titles as a function of user input data.

CALLING SEQUENCE: CALL FNDFLP (DATA, SOL, ALB, PLA, TOT, NV, LC1, LC2, LC3, NTIME, ITIME, TIME, NUNIT)

DATA	- Working storage area where	DATA (1) = SOL (1)
SOL	- Working array	DATA (NTIME+1) = ALB(1)
ALB	- Working array	DATA (2NTIME+1) = PLA(1)
PLA	- Working array	DATA (3NTIME+1) = TOT(1)
TOT	- Working array	
NV	- Number of data items to be plotted	
LC1	- First control character - I (incident) or A (absorbed)	
LC2	- Second control character - F (flux) or R (rate)	
LC3	- Third control character - S (solar), A (albedo), P (planetary), T (total) and ALL (all)	
NTIME	- Number of orbit positions	
ITIME, TIME	- Doublet array containing time and step number	
NUNIT	- Plot output unit	

REFERENCES: PLLOAD

FILES: NSCR1, NSCR2, NOUT

ROUTINE NAME: FRAME

DESCRIPTION: This is a plotting routine to advance to a new frame.

CALLING SEQUENCE: CALL FRAME

REFERENCES: NPMAIN, OPMAIN, PLDRIV



ROUTINE NAME: FRAMEC

DESCRIPTION: This routine determines the number of frames of plot data generated to this point.

CALLING SEQUENCE: CALL FRAMEC (N, I)

N - Number of frames  
I - Dayfile message flag  
= 0 Do not print dayfile message  
= 1 Print dayfile message

REFERENCES: NPMAIN, OPMAIN, PLDRIV

ROUTINE NAME: GBAPRX

DESCRIPTION: This is a user-called routine that computes approximately gray-body factors.

CALLING SEQUENCE: See User's Manual, Appendix D.

REFERENCE: ODPROG (Users Operations Data Block)

FILES: NFF, NGBIR, NGBSO, NOUT, NSCR1, NSCR2, NSCR3

ROUTINE NAME: GBDATA

DESCRIPTION: This is a user-called routine to define the step number and type of gray-bodies to compute.

CALLING SEQUENCE: CALL GBDATA (WBAND, MODFF, NTYP)

NSFF - Step number containing form factors  
WBAND - Waveband to compute gray-bodies for  
2HIR, 3HSOL, and 4HBOTH

REFERENCES: Operations Data Block

ROUTINE NAME: GBEND

DESCRIPTION: This routine can be replaced by the user to intervene  
after completion of the gray-body link.

CALLING SEQUENCE: CALL GBEND

REFERENCES: GBMAIN

ROUTINE NAME: GBHEAD

DESCRIPTION: This routine prints the input parameters defined by  
the user, together with input options.

CALLING SEQUENCE: CALL GBHEAD

REFERENCES: GBMAIN

ROUTINE NAME: GBINV

DESCRIPTION: This routine computes the inverse of the modified form factor matrix stored either in core or in large blocks on disk

CALLING SEQUENCE: CALL GBINV (A, B)

A - Working block 1

B - Working block 2

REFERENCES: GBSCFA

ROUTINE NAME: GBMAIN

DESCRIPTION: This is the main controlling routine in the gray-body link that directs main control and defines the block size.

CALLING SEQUENCE: CALL GBMAIN

REFERENCES: GBPROG

ROUTINE NAME: GBPRE

DESCRIPTION: This routine can be replaced by the user to intervene prior to the call to the invert routine.

CALLING SEQUENCE: CALL GBPRE

REFERENCES: GBMAIN

ROUTINE NAME: GBPROG

DESCRIPTION: This is a preprocessor-generated routine that defines the necessary commons and calls into the main gray-body logic.

CALLING SEQUENCE: CALL GBPROG

REFERENCES: TRASYS (root segment)

ROUTINE NAME: GBRSI

DESCRIPTION: This routine reads gray-body factor data from the Master restart tape and writes it to the NSCR1 file and the master restart output tape.

CALLING SEQUENCE: CALL GBR SI (TYPE, ICOMP)

TYPE - type of gray-body factors on NR SI  
(solar, infrared)  
ICOMP - compute/no compute flag

REFERENCE: GBMAIN

FILES: NR SI, NR SO, NSCR1, NGBIR, NGBSO, NOUT

ROUTINE NAME: GBRTI

DESCRIPTION: This routine reads gray body factor data from the temporary restart tape and writes it to a scratch file and the temporary restart output file.

CALLING SEQUENCE: CALL GBRTI (IGO, A)

IGO - flag indicating stage reached in matrix  
inversion before restart  
A - storage array

REFERENCE: GBINV

FILES: NRTI, NR TO, NSCR1, NSCR2, NSCR3, NOUT

ROUTINE NAME: GBSCFA

DESCRIPTION: Given the set of form factors, this routine blocks the matrix and applies the proper factors to generate a positive definite matrix, and guarantees that the inverse exists.

CALLING SEQUENCE: CALL GBSCFA

REFERENCES: GBMAIN

FILES: Reads NFF, writes NSCR1  
          NGBIR, NGBSO, NOUT, NRSO, NSCR3

ROUTINE NAME: IACT

DESCRIPTION: This routine returns the number of elements in an array defined in the array data block. The integer count of the array is stored in the zeroth cell of the array.

CALLING SEQUENCE: Function IACT (IA)

          IA - First data word of array

REFERENCES: PLDATA, RCDATA

ROUTINE NAME: INITF

DESCRIPTION: This routine is a series of entry points that allow the user to rewind and reuse data files as desired.

CALLING SEQUENCES: CALL DIINIT  
CALL FFINIT  
CALL  
CALL AQINIT  
CALL QOINIT

REFERENCE: ODPROG (Users Operations Data)

FILES: NDI, NFF, NGBSO, NGBIR, NTQ, NOUT

ROUTINE NAME: INIT28

DESCRIPTION: This routine initializes the plot link and sets up the plot file.

CALLING SEQUENCE: CALL INIT28

REFERENCES: PLMAIN, NPMAIN, OPMAIN

ROUTINE NAME: INTOD

DESCRIPTION: This routine sets up the NTITLE array used in page headings and is called each time the operations data are called after a link call.

CALLING SEQUENCE: CALL INTOD

REFERENCES: OPPROG (generated by preprocessor)

ROUTINE NAME: JUPIDD

DESCRIPTION: This routine sets up the planet parameters concerning the planet Jupiter.

CALLING SEQUENCE: CALL JUPIDD

KEY VARIABLES: PRAD - Planet radius  
SOL - Solar constant  
PALB - Planet albedo factor  
WDS - Planet darkside emissive power  
WSS - Planet subsolar emissive power  
GRAV - Gravitational constant

REFERENCES: ORBIT1, ORBIT2, DITTF

ROUTINE NAME: LINE

DESCRIPTION: This routine connects the points  $P_1$  and  $P_2$  defined by  
 $P_1 = (X_1, Y_1)$ ,  $P_2 = (X_2, Y_2)$

CALLING SEQUENCE: CALL LINE (X1, Y1, X2, Y2)

X1, Y1 - Coordinates of point 1  
X2, Y2 - Coordinates of point 2

REFERENCES: NPFPLT, OPFPLT, PLGRID

ROUTINE NAME: LINEOP

DESCRIPTION: This routine defines the intensity of the lines drawn by the plot routines.

CALLING SEQUENCE: CALL LINEOP (DUM, INTEN)

DUM - Dummy parameter  
INTEN - Variable defining intensity

REFERENCES: NPSCAL, OPSCAL

ROUTINE NAME: LINES

DESCRIPTION: This routine connects the arrays of X and Y with a line.

CALLING SEQUENCE: CALL LINES (X, Y, N)

X - Array of X  
Y - Array of Y  
N - Number of points in X and Y arrays

REFERENCES: PLDRIV

ROUTINE NAME: LIST

DESCRIPTION: This routine allows the user to print and line-number data written on files NUSER1 and NBCDOU by the FF, DI, RC and QO segments.

CALLING SEQUENCE: CALL LIST (NF, IEOF)

NF - name of file to be listed  
IEOF - number of end-of-files to be passed during list operation

REFERENCE: ODPROG (Users Operations Data)

FILES: NUSER1, NBCDOU, NOUT



ROUTINE NAME: MAP

DESCRIPTION: This routine maps the plot object space into subject space.

CALLING SEQUENCE: CALL MAP (XMIN, YMIN, XMAX, YMAX, XMI, YMA, YMI, YMA)

XMIN, YMIN, XMAX, YMAX - Corner points of object space

XMI, XMA, YMI, YMA - New corner points of subject space

REFERENCES: NPSCAL, NPINFO, OPSCAL, OPINFO, PLGRID

ROUTINE NAME: MARSD

DESCRIPTION: This routine defines the planet parameters concerning the planet Mars.

CALLING SEQUENCE: CALL MARSD

KEY VARIABLES: PRAD - Planet radius  
SOL - Solar Constant  
PALB - Planet albedo factor  
WDS - Planet darkside emissive power  
WSS - Planet subsolar emissive power  
GRAV - Gravitational constant

REFERENCES: ORBIT1, ORBIT2, DITTP

ROUTINE NAME: MERCUD

DESCRIPTION: This routine sets up the planet parameters concerning the planet Mercury.

CALLING SEQUENCE: CALL MERCUD

KEY VARIABLES: PRAD - Planet radius  
SOL - Solar constant  
PALB - Planet albedo factor  
WDS - Darkside emissive power  
WSS - Subsolar emissive power  
GRAV - Gravitational constant

REFERENCES: ORBIT1, ORBIT2, DITTP

REV 1

III-79

ROUTINE NAME: MODAR

DESCRIPTION: This routine is called by the user in order to change the value of node areas.

CALLING SEQUENCE: See Users Manual, Appendix D.

REFERENCES: ODPROG (Users Operations Data Block)

ROUTINE NAME: MODPR

DESCRIPTION: This routine is called by the user in order to change the values of diffuse nodal surface properties.

CALLING SEQUENCE: See Users Manual, Appendix D.

REFERENCES: ODPROG (Users Operations Data)  
CHGPRP

ROUTINE NAME: MODPRS

DESCRIPTION: This routine is called by the user in order to change the values of specular surface properties.

CALLING SEQUENCE: See Users Manual, Appendix D.

REFERENCES: ODPROG (User Operations Data)  
CHGPRP

FILES: NRAN (random access file)  
NOUT

ROUTINE NAME: MODSHD

DESCRIPTION: This routine is called by the user in order to modify the SHADE/BSHADE flags for surfaces.

CALLING SEQUENCE: See Users Manual, Appendix D.

REFERENCES: ODPROG (Users Operations Data)

FILES: NRAN, NOUT

ROUTINE NAME: MODTR

DESCRIPTION: This routine is called by the user in order to change the values of nodal surface transmissivities.

CALLING SEQUENCE: See Users Manual, Appendix D.

REFERENCES: ODPROG (Users Operations Data)

FILES: NRAN, NOUT

ROUTINE NAME: MOOND

DESCRIPTION: This routine defines the planet parameters concerning the moon.

CALLING SEQUENCE: CALL MOOND

KEY VARIABLES: PRAD - Planet radius  
SOI - Solar constant  
PALB - Albedo factor  
WDS - Darkside emissive power  
WSS - Subsolar emissive power  
GRAV - Gravitational constant

REFERENCES: ORBIT1, ORBIT2, DITTP

ROUTINE NAME: NDATA

DESCRIPTION: This is a user routine to preset the node plot optional parameters.

CALLING SEQUENCE: CALL NDATA (NV, IVU, SCL, ISELN, ITIT, NPHI, NPSI, NOMI, PHI, PSI, OMI)

KEY VARIABLES: See users manual Appendix D.

REFERENCES: User-defined call in the Operations Data block

ROUTINE NAME: NDATAS

DESCRIPTION: This routine is the short-form, user-callable routine to define the node plotter options the user wishes to use.

CALLING SEQUENCE: CALL NDATAS (NV, IVU, SCL)

NV - Plot frame number ( $1 \leq NV \leq 6$ )  
IVU - Plot view type (X, Y, Z, 3-D, GEN)  
SU - Plot scale number

REFERENCES: User-called in the Operations Data block.

ROUTINE NAME: NDUPCK

DESCRIPTION: This routine checks to see if duplicate node numbers are present in the node array.

CALLING SEQUENCE: CALL NDUPCK

REFERENCES: BUILDG

ROUTINE NAME: NEPTD

DESCRIPTION: This routine defines the planet parameters concerning the planet Neptune.

CALLING SEQUENCE: CALL NEPTD

KEY VARIABLES: PRAD - Planet radius  
SOL - Solar constant  
PALE - Planet albedo factor  
WDS - Darkside emissive power  
WSS - Planet subsolar emissive power  
GRAV - Gravitational constant

REFERENCES: ORBIT1, ORBIT2, DITTP

ROUTINE NAME: NODDAT

DESCRIPTION: This routine is called by the user in order to print the current values of all nodal surface properties and their areas.

CALLING SEQUENCE: See Users Manual, Appendix D.

REFERENCES: ODPROG (Users Operations Data Block

FILES: NOUT

ROUTINE NAME: NPAXES

DESCRIPTION: This routine defines the size and orientation of all characters to be small and horizontal, and draws and labels the CCS axes.

CALLING SEQUENCE: CALL NPAXES

KEY VARIABLES: VROT (3, 3) - Direction cosine matrix converting points in the plotter reference coordinate system to points in the desired view.

REFERENCES: NPMAIN

ROUTINE NAME: NPCONE

DESCRIPTION: This routine, given the surface/node dimensions in the SCS generates the portion of the cone defined, converts it to the CCS, and plots the results.

CALLING SEQUENCE: CALL NPCONE

REFERENCES: NPMAIN

ROUTINE NAME: NPCONV

DESCRIPTION: This routine converts the plot control data from common and checks for errors. If no data are defined in common, it sets up the proper default parameters.

CALLING SEQUENCE: CALL NPCONV (NV, NNP, NEND)

NV - Plot view frame number  
NNP - Number of nodes to be selectively plotted  
NEND - End flag determining if more plots are needed

REFERENCES: NPMAIN

ROUTINE NAME: NPCYLO

DESCRIPTION: This routine, given the surface/node dimensions in the SCS, determines the proper plot calls to draw a cylinder.

CALLING SEQUENCE: CALL NPCYLO

REFERENCES: NPMAIN

ROUTINE NAME: NPDISC

DESCRIPTION: This routine, given the surface/node dimensions in the SCS, determines the proper plot calls to draw a disk.

CALLING SEQUENCE: CALL NPDISC

REFERENCES: NPMAIN

ROUTINE NAME: NPDOTL

DESCRIPTION: This routine connects P1 and P2 with a dotted line.

CALLING SEQUENCE: CALL NPDOTL (X1, Y1, Z1, X2, Y2, Z2)

X1, Y1, Z1 - Coordinates of P1  
X2, Y2, Z2 - Coordinates of P2

REFERENCES: NPCONE, NPCYLO, NPPARA, NPSPHE

ROUTINE NAME: NPFPLT

DESCRIPTION: This routine directs the actions of the pen/beam plotter. I is either even or odd, or negative or positive.

CALLING SEQUENCE: CALL NPFPLT (I, X, Y)

I     - Plotting designator  
      - Even, [Draw to (X, Y)]  
      - Odd, [Position at (X, Y)]  
      - Positive, (generates new origin)  
      - Negative, (keeps same origin)  
X, Y - Coordinates of new point

REFERENCES: NPAXES, NPCONE, NPCYLO, NPDOTL, NPTPLT, NPDISC, NPPARA, NPRECT, NPSPHE, NPTRAP

ROUTINE NAME: NPINFO

DESCRIPTION: This routine labels the plot.

CALLING SEQUENCE: CALL NPINFO (NV, KEND, KCC)

NV - View number  
KEND - = 0 Good plot  
       $\neq$  0 Error in plot  
KCC - Internal view number

REFERENCES: NPMAIN

ROUTINE NAME: NPMAIN

DESCRIPTION: This routine is the main driving logic to control the node plot link.

CALLING SEQUENCE: CALL NPMAIN

REFERENCES: NPPROG (preprocessor-generated)

ROUTINE NAME: NPPARA

DESCRIPTION: This routine, given the node dimensions in the SCS, generates the plot calls necessary to draw a paraboloid.

CALLING SEQUENCE: CALL NPPARA

REFERENCES: NPMAIN

ROUTINE NAME: NPPROG

DESCRIPTION: This routine is generated by the preprocessor. It defines the necessary labeled common blocks and calls into the main driving logic for the node plotter.

CALLING SEQUENCE: CALL NPMAIN

REFERENCES: TRASYS (root segment)



ROUTINE NAME: NPRECT

DESCRIPTION: This routine, given the nodal dimensions, generates the plot calls to draw a rectangle.

CALLING SEQUENCE: CALL NPRECT

REFERENCES: NPMAN

ROUTINE NAME: NPRIDA

DESCRIPTION: This routine is unique to the Univac version, and defines a system-labeled common block with a user name and location for labeling the plots. An initialization routine is called from this routine to initialize the plot.

CALLING SEQUENCE: CALL NPRIDA (IARRAY)

IARRAY (1) - Name (word 1)  
IARRAY (2) - Name (word 2)  
IARRAY (3) - User mail number  
IARRAY (4) - Extension  
IARRAY (5) - Blank  
IARRAY (6) - Project

REFERENCES: User-callable routine from the Operations Data block

ROUTINE NAME: NPRNT

DESCRIPTION: This routine prints a summary of node information as the configuration is generated by routines BUILD/ADD.

CALLING SEQUENCE: CALL NPRNT (NRMAS, NAME)

NRMAS - Array of node data  
NAME - Block coordinate system name

REFERENCES: BUILD

ROUTINE NAME: NPROTA

DESCRIPTION: This routine defines the transformation of the direction cosine matrix, given the desired view.

CALLING SEQUENCE: CALL NPROTA

KEY VARIABLES: KC - Desired view number  
PH, PS, OM - Angles necessary to arrive at desired view

REFERENCES: NPMIN

ROUTINE NAME: NPSCAL

DESCRIPTION: This routine determines the grid subject space, maps into the new space, and locates the pen/beam at the new origin.

CALLING SEQUENCE: CALL NPSCAL (XS, YS, X, Y)

XS - Scale factor  
YS - Scale factor  
X - 0.0  
Y - 0.0

REFERENCES: NPAXES

ROUTINE NAME: NPSPE

DESCRIPTION: This routine, given the nodal dimensions, generates the plot calls necessary to draw a sphere or segment of a sphere.

CALLING SEQUENCE: CALL NPSPE

REFERENCES: NPMIN

REV 1  
III-88

ROUTINE NAME: NPTPLT

DESCRIPTION: This routine transforms a point in 3-D space to X, Y coordinates in the subject space, checks to determine if they are in the allowable range, and moves the pen/beam to the coordinates of that point.

CALLING SEQUENCE: CALL NPTPLT (I, XP, YP, X3, Y3, Z3)

I                - Pen-up, pen-down flag  
XP, YP          - Coordinates of transformed point  
X3, Y3, Z3      - Coordinates of input point

REFERENCES: NPCONE, NPCYLO, NPDISC, NPPARA, NPRECT, NPSPHE, NPTRAP, NPMAIN

ROUTINE NAME: NPTRAP

DESCRIPTION: This routine, given the nodal dimensions in the SCS generates the plot calls necessary to draw a trapezoid.

CALLING SEQUENCE: CALL NPTRAP

REFERENCES: NPMAIN

ROUTINE NAME: NUMBER

DESCRIPTION: This routine, given a value and a format, generates a number on the plot frame.

CALLING SEQUENCE: CALL NUMBER (X, F)

X - Floating or integer number to be output  
F - Output format

REFERENCES: NPINFO, OPINFO, PLDRIV, PLGRID

ROUTINE NAME: ODATA

DESCRIPTION: This routine is user-called to define orbit data plotter options.

CALLING SEQUENCE: CALL ODATA (NV, SCL, SCLR, RPLN, TRUE, TIMEP, ISELN, ITIT, NPHF, NPSI, NOMI, OMI, PSI, PHI)

KEY VARIABLES: See Appendix D of users manual.

REFERENCES: User-called in the Operations Data block

ROUTINE NAME: OPAXES

DESCRIPTION: This routine defines the size and orientation of all characters to be small and horizontal, and draws and labels the CCS axis and sun line.

CALLING SEQUENCE: CALL OPAXES

REFERENCES: OPMAIN, OPPLAN

ROUTINE NAME: OPCOMB

DESCRIPTION: This routine does matrix multiplication of the A and B input matrices and returns the results in matrix C.

CALLING SEQUENCE: CALL OPCOMB (A, B, C)

A - Input 3 x 3 matrix  
B - Input 3 x 3 matrix  
C - Resultant matrix ( $C = A*B$ )

REFERENCES: OPLOC, OPLOC2

ROUTINE NAME: OPCONE

DESCRIPTION: This routine, given the surface dimensions in the SCS, generates the portion of the cone defined, converts it to the CCS, and plots the results.

CALLING SEQUENCE: CALL OPCONE

REFERENCES: OPMAIN

ROUTINE NAME: OPCONV

DESCRIPTION: This routine converts the plot control data from common and checks for errors. If no data are defined in common, it sets up the proper default parameters.

CALLING SEQUENCE: CALL OPCONV (NV, NNP, KEND)

NV - Plot view frame number  
NNP - Number of nodes to be selectively plotted  
KEND - End flag determining if more plots are needed

REFERENCES: OPMAIN

ROUTINE NAME: OPCYLO

DESCRIPTION: This routine, given the surface/nodal dimensions in the SCS, determines the proper plot calls to draw a cylinder.

CALLING SEQUENCE: CALL OPCYLO

REFERENCES: OPMAIN

ROUTINE NAME: OPDISC

DESCRIPTION: This routine, given the surface/nodal dimensions in the SCS- determines the proper plot calls to draw a disk.

CALLING SEQUENCE: CALL OPDISC

REFERENCES: OPMAIN

ROUTINE NAME: OPDOTL

DESCRIPTION: This routine connects P1 and P2 with a dotted line.

CALLING SEQUENCE: CALL OPDOTL (X1, Y1, Z1, X2, Y2, Z2)

X1, Y1, Z1 - Coordinates of P1  
X2, Y2, Z2 - Coordinates of P2

REFERENCES: OPCONE, OPCYLO, OPPARA, OPSPHE

ROUTINE NAME: OPEDOT

DESCRIPTION: This routine generates the dotted lines used in the planet- and planet-shadow generation routines and connects points P1 and P2.

CALLING SEQUENCE: CALL OPEDOT (X1, Y1, Z1, X2, Y2, Z2, A)

X1, Y1, Z1 - Coordinates of P1

X2, Y2, Z2 - Coordinates of P2

A - Factor determining the length of the line increment

REFERENCES: OPPLAN, OPSHAD

ROUTINE NAME: OPFPLT

DESCRIPTION: This routine directs the actions of the plot pen/beam. I may be even or odd, negative or positive. The value of I directs the plotter to the coordinates X, Y with the pen up or down.

CALLING SEQUENCE: CALL OPFPLT (I, X, Y)

I - Plot designator

= Even, draw to (X, Y)

= Odd position

= Positive (generates new origin)

= Negative (keep same origin)

REFERENCES: OPAXES, OPCONE, OPDOTL, OPTPLT, OPDISC, OPPARA, OPRECT, OPEDOT, OPSPHE, OPTRAP, OPVCS, OPCYLO

ROUTINE NAME: OPINFO

DESCRIPTION: This routine labels all plot alphabetic data on the plot frame.

CALLING SEQUENCE: CALL OPINFO (NV, KEND)

NV - View frame number

KEND - Error flag

REFERENCES: OPMAIN

ROUTINE NAME: OPLOC

DESCRIPTION: This routine, given the orbital parameters, determines the proper direction cosine matrix to give the proper orientation.

CALLING SEQUENCE: CALL OPLOC

REFERENCES: OPMAIN

REV 1

III-92

ROUTINE NAME: OPLOG2

DESCRIPTION: This routine locates the vehicle relative to the sun and planet according to parameters defined in subroutine DIDT2.

CALLING SEQUENCE: CALL OPLOG2

REFERENCES OPLOC

ROUTINE NAME: OPMAIN

DESCRIPTION: This is the main driving logic to control the orbit plot link.

CALLING SEQUENCE: CALL OPMAIN

REFERENCES: OPPROG (preprocessor-generated)

ROUTINE NAME: OPMAX

DESCRIPTION: This routine determines the maximum radius needed to enclose any surface. This value is then used to determine the proper scale factor.

CALLING SEQUENCE: SCL - OPMAX (NNP)

NNP - Number of surfaces to be selectively plotted

REFERENCES: OPMAIN, OPCONV

ROUTINE NAME: OPPARA

DESCRIPTION: This routine, given the surface dimensions in the SCS, generates the plot calls necessary to draw a paraboloid.

CALLING SEQUENCE: CALL OPPARA

REFERENCES: OPMAIN

ROUTINE NAME: OPPLAN

DESCRIPTION: This routine, given the desired planet radius for plotting (in inches), generates the logic necessary to draw the planet.

CALLING SEQUENCE: CALL OPPLAN

REFERENCES: OPMAIN

ROUTINE NAME: OPPRNT

DESCRIPTION: This routine generates a summary table of orbital parameters on the output file.

CALLING SEQUENCE: CALL OPPRNT

REFERENCES: OPMAIN

ROUTINE NAME: OPPROC

DESCRIPTION: This routine is generated by the preprocessor. It defines the necessary labeled common blocks and the calls into the main driving logic.

CALLING SEQUENCE: CALL OPPROC

REFERENCES: TRASYS (root segment)

ROUTINE NAME: OPRECT

DESCRIPTION: This routine, given the surface dimensions, generates the plot calls to draw a rectangle.

CALLING SEQUENCE: CALL OPRECT

REFERENCES: OPMAIN



ROUTINE NAME: OPROTA

DESCRIPTION: This routine defines the transformed direction cosine matrix, given the desired view.

CALLING SEQUENCE: CALL OPROTA

KEY VARIABLES: KC - Desired view number  
PH, PS, OM - Angles necessary to arrive at desired view

REFERENCES: OPMAIN

ROUTINE NAME: OPSCAL

DESCRIPTION: This routine determines the grid subject space, maps into the new space, and locates the pen/beam at the new origin.

CALLING SEQUENCE: CALL OPSCAL (XS, YS, X, Y)

XS - Scale factor  
YS - Scale factor  
X = 0  
Y = 0

REFERENCES: OPAXES, OPMAIN

ROUTINE NAME: OPSHAD

DESCRIPTION: This routine draws the planet shadow.

CALLING SEQUENCE: CALL OPSHAD

REFERENCES: OPMAIN

ROUTINE NAME: OPSPHE

DESCRIPTION: This routine, given the surface dimensions, generates the plot calls necessary to draw a sphere or segment of a sphere.

CALLING SEQUENCE: CALL OPSPHE

REFERENCES: OPMAIN

ROUTINE NAME: OPTIME

DESCRIPTION: This routine, given the orbit eccentricity, present orbital time and orbital period, computes the true anomaly

CALLING SEQUENCE: CALL OPTIME (TIME, PER, ECC, TRU)

TIME - Input orbital time  
PER - Orbit period  
ECC - Orbit eccentricity  
TRU - Output true anomaly

REFERENCES: OPLOC

ROUTINE NAME: OPTPLT

DESCRIPTION: This routine transforms a point in 3-D space to X, Y coordinates in the subject space, checks to determine if the points are in the allowable range, and moves the pen/beam to the coordinates of that point.

CALLING SEQUENCE: CALL OPTPLT (I, XP, YP, X3, Y3, Z3)

I - Pen-up, pen-down flag  
XP, YP - Coordinates of transformed point  
X3, Y3, Z3 - Coordinates of input point

REFERENCES: OPCONE, OPCYLO, OPDISC, OP PARA, OPSPHE, OPTRAP, OPRECT

ROUTINE NAME: OPTRAP

DESCRIPTION: This routine, given the surface dimensions in the SCS, generates the plot calls necessary to draw a trapezoid.

CALLING SEQUENCE: CALL OPTRAP

REFERENCES: OPMAIN

ROUTINE NAME: OPTRNP

DESCRIPTION: This routine, given a 3 x 3 matrix, computes the transpose.

CALLING SEQUENCE: CALL OPTRNP (A, B)

A - 3 x 3 matrix to be transposed

B - 3 x 3 resultant matrix

REFERENCES: OPDOTL, OPTPLT, OPLOC, OPLOC2

ROUTINE NAME: OPTRS3

DESCRIPTION: This routine changes a translation vector from one coordinate system to the corresponding translation vector in the new system.

CALLING SEQUENCE: CALL OPTRS3 (X, Y, Z, A, B, C, RX, RY, RZ, TRAN)

X, Y, Z - New translation vector

A, B, C - Translation vector in the old system

RX, RY, RZ - Translation vector in the new system

TRAN - Direction cosines relating the old system to the new system

REFERENCES: OPAXES, OPDOTL, OPTPLT, OPEDOT, OPVCS

ROUTINE NAME: OPUNIT

DESCRIPTION: This routine generates an identity matrix.

CALLING SEQUENCE: CALL OPUNIT (A)

A - Output unit matrix

REFERENCES: OPLOC, OPLOC2

ROUTINE NAME: OPVCS

DESCRIPTION: This routine labels the vehicle axes on the plot.

CALLING SEQUENCE: CALL OPVCS

REFERENCES: OPMAIN

ROUTINE NAME: ORBIT1

DESCRIPTION: This routine can be called by the user in the Operations Data block to define the orbit.

CALLING SEQUENCE: CALL ORBIT1 (PLANAM, ALANI, APEI, OIN, TIME, HPI, HAI, SRA, SDE, STA, STD)

KEY VARIABLES: See users manual, Appendix D

REFERENCES: ODPROG

ROUTINE NAME: ORBIT2

DESCRIPTION: This is a user callable routine to define the desired orbit.

CALLING SEQUENCE: CALL ORBIT2 (PLANAM, CIG, BET, CIGS, BETS, TIME, HPI, HAI)

KEY VARIABLES: See users manual, Appendix D

REFERENCES: ODPROG

ROUTINE NAME: ORIENT

DESCRIPTION: This routine allows the user to define the vehicle orientation.

CALLING SEQUENCE: CALL ORIENT (TYPE, IROTX, IROTY, IROTZ, ROTX, ROTY, ROTZ)

KEY VARIABLES: See users manual, Appendix D

REFERENCES: ODPROG

ROUTINE NAME: PAGE

DESCRIPTION: This routine is called prior to every written statement to the output file to count the line printed and take care of all paging.

CALLING SEQUENCE: II = PAGE (I)  
                  II = 0 New page was written  
                  1 No new page  
                  I - Number of lines to be printed

REFERENCES: All output generating routines.

ROUTINE NAME: PLCMBN

DESCRIPTION: This routine combines incident and absorbed flux data according to correspondence data prior to data plotting.

CALLING SEQUENCE: CALL PLCMBN (SOL, ALB, PLA, NDCOM, NNDC, NUNIT)

SOL - Solar flux/rate array  
ALB - Albedo flux/rate array  
PLA - Planetary flux/rate array  
NOCOM - Combine/no combine flag  
NNDC - Number of nodes after combining  
NUNIT - flux/rate data storage file name (NDI, NTQ)

REFERENCES: FNDFLP

FILES: NTQ, NOUT, NGBIRR

ROUTINE NAME: PLCFIT

DESCRIPTION: This routine does smooth-curve fitting. Given two arrays LO words long (X, Y), it generates NO points between each set of X, Y points and stores the points in arrays U and V.

CALLING SEQUENCE: CALL PLCFIT (LO, X, Y, NO, U, V)

LO - Number of input points  
X - Independent variable array  
Y - Dependent variable array  
NO - Number of divisions between each set of points  
U, V - Output arrays

REFERENCES: PLDCON

ROUTINE NAME: PLDATA

DESCRIPTION: This routine can be called by the user in the Operations Data block to define PLOT link options.

CALLING SEQUENCE: CALL PLDATA (IP, INS, IS, CRVF, TLX, TLY, T1, T2, XMPF, YMPF)

KEY VARIABLES: See users manual, Appendix D

REFERENCES: Operations Data block

ROUTINE NAME: PLDCON

DESCRIPTION: This routine checks for discontinuities in the output plot and calls for curve fitting between all discontinuities.

CALLING SEQUENCE: CALL PLDCON (NTIME, TIME, PARRAY, NODIV, NTOT, TIMP, PLOTP)

NTIME - Number of orbit points  
TIME - Array of times  
PARRAY - Array of dependent data  
NODIV - Number of divisions between each point  
NTOT - Number of output points  
TIMP - Array of curve-fit points (independent)  
PLOTP - Array of curve-fit points (dependent)

REFERENCES: PLDRIV

ROUTINE NAME: PLDRIV

DESCRIPTION: This routine is the main driving routine in the plot link. It decodes the type of plot and calls the proper routines.

CALLING SEQUENCE: CALL PLDRIV (RINDEP, S, DEPEND, RINDO, DEPO, IS, NDIV, NINDV)

RINDEP - Independent variable array  
S - Temporary working array  
DEPEND - Dependent variable array  
RINDO - Independent variable array computed  
DEPO - Dependent variable array output  
IS - Temporary array (same as S)  
NDIV - Number of divisions between curve-fit points  
NINDV - Number of independent variable points allowed

REFERENCES: PLMAIN

ROUTINE NAME: PLGRID

DESCRIPTION: This routine draws the plot grid and labels the complete frame.

CALLING SEQUENCE: CALL PLGRID (XMIN, XMAX, YMIN, YMAX, NODEN)

XMIN, XMAX, YMIN, YMAX - Minimum and maximum dimensions of  
plot frame  
NODEN - Node number of current frame

REFERENCES: PLDRIV

ROUTINE NAME: PLLOAD

DESCRIPTION: This routine determines the type of plot and finds the plot data if absorbed or incident flux data are to be plotted.

CALLING SEQUENCE: CALL PLOAD (DATA, TIME, ITIME, NTIME, NINDV)

DATA	- Working temporary array
TIME, ITIME	- Doublet array containing time and step number
NTIME	- Number of time points
NINDV	- Number of independent-variable points

REFERENCES: PLMAIN

ROUTINE NAME: PLMAIN

DESCRIPTION: This is the main driving routine in the plot segment. Its primary functions are to determine the size of the array and allocate array space.

CALLING SEQUENCE: CALL PLMAIN

REFERENCES: PLPROG (preprocessor-generated)

ROUTINE NAME: PLOUT

DESCRIPTION: This routine applies the dependent variable multiplier and converts fluxes to rates.

CALLING SEQUENCE: CALL PLOUT (NV, A1, A2, A3, A4, ITYPE)

NV	- Number of arrays
A1, A2, A3, A4	- Data arrays
ITYPE	- Flag indicating flux or rate
	= 4HFLUX heat flux
	= 4HRATE heat rate

REFERENCES: FNDFLP

ROUTINE NAME: PLPROG

DESCRIPTION: This routine is generated by the preprocessor and calls in the main processor routines.

CALLING SEQUENCE: CALL PLPROG

REFERENCES: TRASYS (root segment)

ROUTINE NAME: PLSCL

DESCRIPTION: This routine, given the maximum and minimum values, determines a scale using convenient units.

CALLING SEQUENCE: CALL PLSCL (BMAX, BMIN, NSQ, AU, AL, S)

BMAX, BMIN - Maximum and minimum values to be plotted  
NSQ - Increment desired  
AU, AL - New upper and lower values  
S - Scale factor

REFERENCES: PLGRID

ROUTINE NAME: PLSYMB

DESCRIPTION: This routine, given an array of Hollerith data, outputs the data to the plot frame and supplies the proper character terminator.

CALLING SEQUENCE: CALL PLSYMB (ARRAY, N)

ARRAY - Array of data to be printed  
N - Number of words in array

REFERENCES: PLGRID



ROUTINE NAME: PLUTOD

DESCRIPTION: This routine defines the planet parameters concerning the planet Pluto.

CALLING SEQUENCE: CALL PLUTOD

KEY VARIABLES: PRAD - Planet radius  
SOL - Solar constant  
PALB - Albedo factor  
WDS - Darkside emissive power  
WSS - Subsolar emissive power  
GRAV - Gravitational constant

REFERENCES: ORBIT1, ORBIT2, DITTP

ROUTINE NAME: PRDUMP

DESCRIPTION: This routine writes the node, area and property arrays to the RSO file.

CALLING SEQUENCE: CALL PRDUMP (LABEL)

LABEL - Flag indicating whether properties written out were defined per surface data block, as modified by "MOD" routines or are combined over correspondence data.

REFERENCES: ODPROG, GMMAIN

FILES: NRSO, NOUT

ROUTINE NAME: QOAVGR

DESCRIPTION: This routine computes the integrated average of the absorbed heat rates that were input.

CALLING SEQUENCE: CALL QOAVGR (NTIME, TIME, NUNIT)

NTIME - Number of time points

TIME - Doublet array of time and step number

NUNIT - Unit number sorted fluxes heat rates are stored on

REFERENCES: QOMAIN

ROUTINE NAME: QOAVGS

DESCRIPTION: This routine, given the final data, outputs the data according to the user requirements.

CALLING SEQUENCE: CALL QOAVGS (NODNO, QAV, AREAT)

NODNO - Node number

QAV - Averaged Q value

AREAT - Area of node

REFERENCES: QOSAVE

ROUTINE NAME: QOCMBN

DESCRIPTION: This routine generates the combining data read from the correspondence data.

CALLING SEQUENCE: CALL QOCMBN (ICOMBL, IFIRSL)

ICOMBL - Number of points in Combine Array 1

IFIRSL - Number of points in Combine Array 2

REFERENCES: QOMAIN

ROUTINE NAME: QOCOMB

DESCRIPTION: Based on the combine arrays, this routine combines and stores the data. A final call to this routine causes the combined tables to be output

CALLING SEQUENCE: CALL QOCOMB (DATA1, DATA2, ND1M1, ND1M2, ND1M, IFIRSL, ICOMBL, DATA, IROW, KTAB)

DATA1, DATA2	- Working storage blocks
ND1M1, ND1M2	- Length of data blocks 1 and 2
ND1M	- Number of nodes
IFIRSL, ICOMBL	- Length of combining arrays
DATA	- Working storage array
IROW	- Sequence number of current row
KTAB	- Table number

REFERENCES: QOSAVE

ROUTINE NAME: QODATA

DESCRIPTION: This routine is called by the user to define particular parameters for the absorbed heat rate output.

CALLING SEQUENCE: CALL QODATA (IARRAY, NTMARY, QOTP, SUN, AMPF, FMPF, TMPF, TYPE, NCOR)

KEY VARIABLES: See users manual, Appendix D for definition of variables

REFERENCES: Operations Data block

ROUTINE NAME: QOFLIP

DESCRIPTION: This routine, given a matrix (NTIME x ICOL) stored on disk NS1, converts this to a matrix (ICOL x NTIME) on unit NS2.

CALLING SEQUENCE: CALL QOFLIP (DATA, NTIME, ICOL, NS1, NS2)

DATA	- Working storage area
NTIME	- Number of rows
ICOL	- Number of columns
NS1, NS2	- Input, output unit numbers

REFERENCES: QOMAIN

ROUTINE NAME: QOHEAD

DESCRIPTION: This routine outputs default and user control parameters to the output file.

CALLING SEQUENCE: CALL QOHEAD

REFERENCES: QOMAIN

ROUTINE NAME: QOMAIN

DESCRIPTION: This routine is the main driving logic for the QOCAL link.

CALLING SEQUENCE: CALL QOMAIN

REFERENCES: QOPROG

ROUTINE NAME: QOPROG

DESCRIPTION: This routine is defined by the preprocessor. It defines all needed labeled common blocks and calls in the driving logic for the calculation of absorbed heat rates.

KEY VARIABLES: CALL QOPROG

REFERENCES: TRASYS (root segment)

ROUTINE NAME: QOSAVE

DESCRIPTION: This routine reads the uncombined data points and, from the input options directs the combining and output options.

CALLING SEQUENCE: CALL QOSAVE (ICOMBL, IFIRSL, NTIME, TIME)

ICOMBL, IFIRSL - Length of the combining arrays  
NTIME - Number of time points  
TIME - Double array of step numbers and times

REFERENCES: QOMAIN

ROUTINE NAME: QOSBCD

DESCRIPTION: This routine is called to output the subroutine Call cards.

CALLING SEQUENCE: CALL QOSBCD (PER, ITIME, KTAB, AREAT, NODNO)

PER - Orbit period  
ITIME - Time array number  
KTAB - Table number reference number  
AREAT - Area  
NODNO - Node number

REFERENCES: QOSAVE

ROUTINE NAME: QOTABS

DESCRIPTION: This routine is called to generate the final output tables.

CALLING SEQUENCE: CALL QOTABS (KTAB, DATA, NTIME, ARE)

KTAB - Output array number  
DATA - Arrays of Q data  
NTIME - Number of data points  
ARE - Area of node

REFERENCES: QOSAVE, QOCOMB

ROUTINE NAME: QOTMES

DESCRIPTION: This routine is called to output the time array in final form.

CALLING SEQUENCE: CALL QOTMES (ITIME, TIME, NTIME)

ITIME - Time array number  
TIME - Array of times  
NTIME - Number of time points

REFERENCES: QOSAVE

ROUTINE NAME: RBAREA

DESCRIPTION: This is a function routine that calculates elemental areas for unevenly distributed elements of unequal size.

CALLING SEQUENCE: RBAREA (ILP, ALPH, BMIN, BMAX, GMIN, GMAX)

ILP	-	Surface type
ALPH	-	
BMIN	-	
BMAX	-	
GMIN	-	
GMAX	-	

} Surface dimensions of current element

REFERENCES: RBEXPN, RBXPNI

ROUTINE NAME: RBCAL

DESCRIPTION: This routine contains the driving logic to determine the number and distribution of elements on node IN, as well as on the image of node JN as seen in specular surface MIRROR. It also calculates the image factor from node IN to node JN.

CALLING SEQUENCE: CALL RBCAL (MIRROR)

MIRROR - Surface sequence number of current specular surface

KEY VARIABLES: IN, JN - Sequence numbers of current node pair  
FE, FA - IR and solar form factors from node IN to image of node JN in specular surface MIRROR  
WE, WA - IR and solar shadowing factors  
RBVALI, RBVALS - IR and solar image factors between node IN and node JN  
KN - Sequence number of current specular node  
SREFLI, SREFLS - Arrays of IR and solar specular reflectance values

REFERENCES: RBPPE

FILES: NOUT - System output file

ROUTINE NAME: RBCALI

DESCRIPTION: This routine is analogous to RBCAL. It is called when the I - J order of the nodes was reversed due to area differences.

CALLING SEQUENCE: See RBCAL

REFERENCES: See RBCAL

FILES: See RBCAL

ROUTINE NAME: RBDATA

DESCRIPTION: This routine is called by the user in order to define parameters necessary to execute the RBCAL segment.

CALLING SEQUENCE: CALL RBDATA (NFIGFF, FFACC, FFACCS, FFRATL, FFPRT)

NFIGFF - Configuration name for form factor access

FFACC }  
FFACCS } See subroutine FFDATA, Users Manual,  
FFRATL } Appendix D  
FFPRT }

REFERENCES: ODPROG (Users Operations Data)

ROUTINE NAME: RBELEM

DESCRIPTION: Given the number of elements in each direction, this routine calculates the elemental position and area vectors.

CALLING SEQUENCE: CALL RBELEM (NB, NG, ILP, IC, DATA, POS, ARA, TRAN, RX, RY, RZ)

NB, NG - Number of elements in the beta and gamma directions  
ILP - Surface type  
IC - Counter for total number of elements on node pair  
DATA - Array of node dimensions  
POS - Array of elemental position vectors  
ARA - Array of elemental area vectors  
TRAN - Matrix of direction cosines  
RX, RY, RZ - Components of SCS origin position vector in the ICS, BCS, or CCS

REFERENCES: RBCAL, RBEXPN, RBXPNI

ROUTINE NAME: RBELSL

DESCRIPTION: Given the total number of elements required on a node, this routine makes them as square as possible and determines the number of elements in each direction.

CALLING SEQUENCE: CALL RBELSL (NB, NG, ILP, DATA, NTOT)

NB, NG - Number of elements in the beta and gamma directions  
ILP - Surface type  
DATA - Array of node dimensions  
NTOT - Number of elements required on a node

REFERENCES: RBCAL, RBEXPN, RBCALI, RBXPNI



ROUTINE NAME: RBEND

DESCRIPTION: This routine provides for user intervention after the image factor calculation. It normally calls for a time accounting for the problem and end-files the image factor file on NFF.

CALLING SEQUENCE: CALL RBEND

REFERENCES: RBMAIN

FILES: NFF - Disk file for storing image factors.

ROUTINE NAME: RBESUM

DESCRIPTION: This routine provides an accounting, in CP seconds, of the time required to calculate image factors.

CALLING SEQUENCE: CALL RBESUM

REFERENCES: RBEND

FILES: NOUT - System output file

ROUTINE NAME: RBEXP

DESCRIPTION: This routine expands nodes into subnodes, determines the number and distribution of elements on each subnode, and calculates more accurate image factors than would be possible on a nodal basis.

CALLING SEQUENCE: CALL RBEXP (POSI, ARAI, FE, FA, RATI, RATJ, JFLAG, JST, NEI, NEJ, NSUR, IADDR, MIRROR)

POSI	- Array of elemental position vectors
ARAI	- Array of elemental area vectors
FE, FA	- Resultant IR and solar form factors
RATI, RATJ	- Ratio of maximum number of elements to the no. of elements indicated by arithmetic averaging
JFLAG	- Flag indicating that maximum number of elements was exceeded
JST	- Number of possible shadowers
NEI, NEJ	- Number of elements on nodes I and J
NSUR	- Actual number of shadowing surfaces
IADDR	- Maximum time return sequence (not used)
MIRROR	- Surface sequence number of current specular surface

REFERENCES: RBCAL

ROUTINE NAME: RBGTST

DESCRIPTION: This is an integer function routine that tests a given value, GT, to determine if it falls in the range  $GN < GT < GX$ . If it does, the function value is 0 (zero); if not, the function value is 1 (one).

CALLING SEQUENCE: RBGTST (GN, GX, GT)

GN	- Lower bound
GX	- Upper bound
GT	- Value to be tested

REFERENCES: RBSHD

ROUTINE NAME: RBIMAG

DESCRIPTION: This routine images surfaces for shadowing purposes  
and stores the results on NSCR2 for later use.

CALLING SEQUENCE: CALL RBIMAG (NST)

NST - Total number of shadowing surfaces

KEY VARIABLES: JS - Surface sequence number of current  
specular surface  
NSURFS - NST plus number of images in specular  
surface JS  
IFS, IKS, PR,  
DSTR, DIMS,  
PSH, TSTR - Surface description parameters

REFERENCES: RBMAIN

FILES: NSCR2 - Scratch file

ROUTINE NAME: RBMAIN

DESCRIPTION: This routine contains the main driving logic for calculating image factors.

CALLING SEQUENCE: CALL RBMAIN

KEY VARIABLES: IN - Sequence number of "viewer" node  
JN - Sequence number of node whose image in KN  
is viewed by IN  
KN - Sequence number of specular node  
ISPEC - Array of specular surface sequence numbers

REFERENCES: RBPROG (preprocessor-developed)

FILES: NRAN - Random access file  
NSCR2 - Scratch file  
NOUT, NRTI, NRTO

ROUTINE NAME: RBNIMG

DESCRIPTION: This routine images the current node JN in the current specular surface.

CALLING SEQUENCE: CALL RBNIMG (JM, NOIM)

JM - Surface sequence number of current specular surface

NOIM - Flag to indicate if an image of the current node exists in JM

= 0, an image exists

= 1, no image

KEY VARIABLES: POSNJ, RXJ  
RYJ, RZJ, DATAJ, } Nodal description parameters  
TRANJ

REFERENCES: RBMAIN

FILES: NRAN - Random access file

ROUTINE NAME: RBOUT

DESCRIPTION: This is a user routine that can be replaced to change the form of output for image factors. The normal calls are to RBPNCB (punches image factors) and RBPRNT (prints image factors).

CALLING SEQUENCE: CALL RBOUT

REFERENCES: RBMAIN, RBRSI

ROUTINE NAME: RBPOSI

DESCRIPTION: This routine transposes a vector in the ICS, BCS, or CCS into the SCS of the current specular surface, negates the Z component, and transforms the vector back into the ICS, BCS, or CCS.

CALLING SEQUENCE: CALL RBPOSI (X, Y, Z, TRAN)

X, Y, Z - Components of vector in the ICS, BCS, or CCS  
TRAN - Matrix of direction cosines

KEY VARIABLES: A, B, C - Components of vector in the SCS

REFERENCES: RBIMAG, RBNIMG

ROUTINE NAME: RBPRE

DESCRIPTION: This routine provides for user intervention prior to the calculation of an image factor. The normal call is to RBCAL, which calculates the image factor.

CALLING SEQUENCE: CALL RBPRE (MIRROR)

MIRROR - Surface sequence number of current specular surface

REFERENCES: RBMAIN

ROUTINE NAME: RBPRNT

DESCRIPTION: This routine prints the image factor between node IN and node JN. The user can override this routine to change the output format if he desires.

CALLING SEQUENCE: CALL RBPRNT

KEY VARIABLES: IN - Sequence number of "viewer" node  
JN - Sequence number of "viewee" node  
NODE - Array of node numbers  
RBVALI, RBVALS - IR and solar image factors

REFERENCES: RBOU

FILES: NOUT - System output file

ROUTINE NAME: RBPSHD

DESCRIPTION: This routine determines all possible shadowing surfaces between node I and the image of node J in specular surface MIRROR.

CALLING SEQUENCE: CALL RBPSHD (RADJ, RADI, POSJ, POSI, NSURF, NSS, JST, IN, JN, MIRROR)

RADJ, RADI - Radii of sphere enclosing nodes I and J  
POSJ, POSI - Position vectors of enclosing spheres  
NSURF - Number of shadowing surfaces  
NSS - NSURF plus all surface images in specular surface MIRROR  
JST - Number of possible shadowing surfaces found  
IN, JN - Sequence numbers of surfaces containing nodes I and J  
MIRROR - Sequence number of current specular surface

REFERENCES: RBCAL, RBEXPN, RBCALI, RBXPNI

ROUTINE NAME: RBRDIN

DESCRIPTION: This routine sets up the shadowing data arrays, reads form factors from the NFF file written in the FF link and writes them on a scratch file for later use, and initializes the NFF file for storage of image factors.

CALLING SEQUENCE: CALL RBRDIN

KEY VARIABLES: NSPEC - Number of active specular surfaces  
NSURF - Number of shadowing surfaces  
DIMS, TSTR, DST, IFS, IKS, - Shadowing data arrays  
PR, PSB  
ISPEC - Array of specular surface sequence numbers  
SREFLI, SREFLS - IR and solar specular reflectance arrays

REFERENCES: RBMAIN

FILES: NFF - Disk file for storage of image factors  
NOUT - System output file  
NRAN - Random access file  
NSCRI - Scratch file  
NRSI, NRSO

REV 1

III-116

ROUTINE NAME: RBRDRQ

DESCRIPTION: This routine initializes the image factor arrays and  
reads restart values if they are available.

CALLING SEQUENCE: CALL RBRDRQ

KEY VARIABLES: FFVALI, FFVALS - IR and solar form factors from the FF link.  
Also used for image factor restart values  
RBVALI, RBVALS - IR and solar image factor arrays

REFERENCES: RBMAIN

FILES: NSCR1 - Scratch file

ROUTINE NAME: RBROW

DESCRIPTION: This routine writes image factor data on file NFF by rows.

CALLING SEQUENCE: CALL RBROW

KEY VARIABLES: IN - Image factor row number  
NODE - Array of node numbers  
RBVALI, RBVALS - IR and solar image factors

REFERENCES: RBMAIN

FILES: NFF - Disk file for storing image factors  
NRSO

ROUTINE NAME: RBRPSN

DESCRIPTION: This routine calculates the minimum radius of a sphere that will encompass a given subnode and determines the position vector to the center of the sphere.

CALLING SEQUENCE: CALL RBRPSN (RADN, POSN, ILK, DATA, BETA, GAMMA, DB, DG)

RADN - Radius of encompassing sphere  
POSN - Position vector of sphere  
ILK - Surface type of node  
DATA - Surface dimension values  
BETA - Distance from edge of node to center of current subnode in the beta direction  
GAMMA - Distance from edge of node to center of current subnode in the gamma direction  
DB, DG - Dimensions of subnode in the beta and gamma directions

REFERENCES: RBEXPN, RBXPNI



ROUTINE NAME: RBRSI

DESCRIPTION: This routine reads rows of image factors from the  
NRSI file and individual image factors from the NRSI  
file.

CALLING SEQUENCE: CALL RBRSI (ICOMP)

ICOMP - flag indicating whether or not all  
image factors were found on restart  
tapes.

REFERENCES: RBRDIN

FILES: NRSI, NRTI, NRSO, NFF, NOUT, NRTO

ROUTINE NAME: RBRSUM

DESCRIPTION: This routine prints out the time required to calculate  
one row of image factors.

CALLING SEQUENCE: CALL RBRSUM

REFERENCES: RBROW, RBRSI

FILES: NOUT - System output file

ROUTINE NAME: RBSHD

DESCRIPTION: This routine calculates the elemental shadowing between surfaces IN and the image of surface JN in specular surface MIRROR. An element is either completely shadowed or not shadowed at all. Shadowing is reduced by the transmissivity of semitransparent shadowing surfaces.

CALLING SEQUENCE: CALL RBSHD (ILKI, RX, RY, RZ, POSI, WE, WA, JST, IN, JN, I, NSS, RS, MIRROR)

KEY VARIABLES: ILKI - Surface type of IN  
RX, RY, RZ - Components of vector from element on node I to element on node J  
POSI - Array of elemental position vectors  
WE, WA - Elemental shadowing factors for IR and solar image factors  
JST - Number of possible shadowing surfaces  
IN, JN - Surface sequence numbers  
I - Element sequence number  
NSS - Number of possible shadowers (including images)  
RS - Square of the magnitude of the element-to-element connecting vector ( $RX^2 + RY^2 + RZ^2$ )  
MIRROR - Sequence number of current specular surface

REFERENCES: RBCAL, RBEXPN, RBCALI

ROUTINE NAME: RBTMCK

DESCRIPTION: This routine calculates the time remaining in the run and compares it with the estimated time for the next calculation. If insufficient time remains, the run is aborted.

CALLING SEQUENCE: CALL RBTMCK (IADDR, MAXLFT)

IADDR - Not used  
MAXLFT - Estimated time required for next calculation

KEY VARIABLES: M - Time remaining in run

REFERENCES: RBCAL, RBEXPN, RBCALI, RBXPNI

FILES: NOUT - System output file

REV 1

III-120

ROUTINE NAME: RBTRS3

DESCRIPTION: This routine transforms points in an SCS to points in the ICS, BCS, or CCS.

CALLING SEQUENCE: CALL RBTRS3 (X, Y, Z, A, B, C, RX, RY, RZ, TRAN)

X, Y, Z - Coordinates of point in the ICS, BCS, or CCS  
A, B, C - Coordinates of point in the SCS  
RX, RY, RZ - Components of the SCS origin position vector  
in the ICS, BCS, or CCS  
TRAN - Matrix of direction cosines

REFERENCES: RBELEM, RBEXPN, RBSHD, RBXPNI

ROUTINE NAME: RBVIEW

DESCRIPTION: This routine calculates the unshadowed form factor between node IN and the image of node JN. It also calculates RATI and RATJ, which will later be compared with the user-input variable, EFRATL, to see if the nodes should be expanded into subnodes for better accuracy.

CALLING SEQUENCE: CALL RBVIEW (NTI, NTJ, POSI, ARAI, NTOTI, NTOTJ, ARI, ARJ, NOT, FRACT, RATI, RATJ)

NTI - Initial number of elements on node IN  
NTJ - Total initial number of elements on node IN  
and the image of node JN  
POSI - Array of elemental position vectors  
ARAI - Array of elemental area vectors  
NTOTI, NTOTJ - Number of elements required on node IN  
and the image of node JN  
ARI, ARJ - Areas of node IN and the image of node JN  
NOT - Flag to indicate whether node IN can "see"  
the image of node JN  
= 0 Can see  
= 1 Cannot see  
FRACT - Unshadowed node-to-image form factor  
RATI, RATJ - Ratios of indicated maximum elements re-  
quired to NTOTI and NTOTJ

REFERENCES: RBCAL, RBCALI

ROUTINE NAME: RBVWT

DESCRIPTION: This routine calculates unshadowed form factors between subnodes and determines the number of elements required, based on a weighted-average criterion.

CALLING SEQUENCE: CALL RBVWT (NTI, NTJ, POSI, ARAI, NTOTI, NTOTJ, ARI, ARJ, NOT, FRACT)

NTI	- Number of elements on the subnode of node IN
NTJ	- Total number of elements on the subnode of node IN and the subnode of the image of node JN
POSI	- Array of elemental position vectors
ARAI	- Array of elemental area vectors
NTOTI, NTOTJ	- Number of elements required on the subnodes of node IN and the image of node JN
ARI, ARJ	- Areas of subnodes
NOT	- Flag to indicate whether the subnode on node IN can "see" the image of node JN = 0 Can see = 1 Cannot see
FRACT	- Unshadowed subnode-to-subnode form factor

REFERENCES: RBEXPXN, RBXPNI

ROUTINE NAME: RBXPNI

DESCRIPTION: This routine is analogous to subroutine RBEXPXN. It is called when the I - J order of node pairs are reversed due to area differences.

CALLING SEQUENCE: Reference RBEXPXN

REFERENCES: Reference RBEXPXN

FILES: Reference RBEXPXN

ROUTINE NAME: RCCMBN

DESCRIPTION: This routine, given an array defining combinations, combines and calls the output routines.

CALLING SEQUENCE: CALL RCCMBN (ICOMB, ICOMBL, SF, SPACNO, NUNIT)

ICOMB - Array of combination data  
ICOMBL - Length of ICOMB array  
SF - Temporary array to store script F  
SPACNO - Array to store script F to space  
NUNIT - Unit containing gray-body matrix

REFERENCES: RCMAN

FILES: NSCR3, NSCR1, NSCR2

ROUTINE NAME: RCDATA

DESCRIPTION: This routine is user-called in the Operations Data block and defines parameters for the RCCAL link.

CALLING SEQUENCE: CALL RCDATA (NSGBYR, PNCH, FMIN, IRKN, RKSPC, NSPAC, SIG, AMPF, TAPE, RADI, NEFI, IPRIM, ISEC)

KEY VARIABLES: See user's manual, Appendix D

REFERENCES: Operations data block

ROUTINE NAME: RCEND

DESCRIPTION: This routine can be replaced by the user to intervene just prior to the end of the RCCAL link.

CALLING SEQUENCE: CALL RCEND

REFERENCES: RCMAN

ROUTINE NAME: RCHEAD

DESCRIPTION: This routine prints the control parameters on the output file.

CALLING SEQUENCE: CALL RCHAD

REFERENCES: RCMAN

ROUTINE NAME: RCINIT

DESCRIPTION: Combines nodes according to correspondence data.

CALLING SEQUENCE: CALL RCINIT (NOCOM)

NOCOM - flag to indicate if any combining was performed

REFERENCES: RCMAN

FILES: NGBIRR, NOUT

ROUTINE NAME: RCMAN

DESCRIPTION: This is the main driving logic of the RCCAL link and directs the main logic flow.

CALLING SEQUENCE: CALL RCMAN

REFERENCES: RCPRG (preprocessor generated)

ROUTINE NAME: RCPNCH

DESCRIPTION: This routine defines the final output form of the  
RADKS in the form of cards, BCD tape, or binary tape.

CALLING SEQUENCE: CALL RCPNCH (ICN, NI, NJ, SIG, SFA)

ICN - Conductor number  
NI - Node i  
NJ - Node j  
SIG - Stephan-Boltzmann constant  
SFA - Script F area factor

REFERENCES: RCMAN, RCPSS2

FILES: NOUT

ROUTINE NAME: RCPRE

DESCRIPTION: This routine can be replaced by the user to provide  
intervention prior to the RCCAL computation in the  
RCCAL link.

CALLING SEQUENCE: CALL RCPRE

REFERENCES: RCMAN

ROUTINE NAME: RCPROG

DESCRIPTION: This routine is generated by the preprocessor. It  
defines all labeled common blocks required by the  
RCCAL link, as well as calls into the main logic.

CALLING SEQUENCE: CALL RCPROG

REFERENCES: TRASYS (root segment)

ROUTINE NAME: RCPSS1

DESCRIPTION: This routine sorts rows of radiation conductors into decreasing order and sets up special identification flags.

CALLING SEQUENCE: CALL RCPSS1 (NND, SORT, ISORT, IFLG)

NND - number of values in row of radiation conductors

SORT

ISORT arrays used in sorting process

IFLG

REFERENCES: RCMAN

FILES: NSCR2, NSCR3, NOUT

ROUTINE NAME: RCPSS2

DESCRIPTION: This routine condenses the matrix of radiation conductors by saving only significant conductors according to user criteria and punches out significant radiation conductors.

CALLING SEQUENCE: CALL RCPSS2 (NND, ICN, ESUM, IFLG)

NND - Reference RCPSS1

ICN - Radiation conductor number

ESUM - Array of radiation conductor sums

IFLG - Scratch array

REFERENCES: RCMAN

FILES: NSCR2, NSCR3, NOUT



ROUTINE NAME: RCTRAN

DESCRIPTION: This routine transposes the gray body factor matrix.

CALLING SEQUENCE: CALL RCTRAN (A,B,D,E, NB, NN, NR, NRECT,  
ITRC50, MAXBC)

A	}	Scratch arrays
B		
D		
E		
NB, NR - Size flags		
NN - Number of nodes		
NRECT - Record count flag		
ITRC50 - Trace flag		
MAXBC - Amount of core available for transposing operation		

REFERENCES: RCMAN

FILES: NSCR2, NGBIR, NOUT, NSCR1

ROUTINE NAME: RDMAIN

DESCRIPTION: This routine initializes the variables used in the  
processor via three methods:

- 1) Reads in the sequential file;
- 2) Sets variables with predefined data;
- 3) Reads in the random I/O file.

CALLING SEQUENCE: CALL RDMAIN

REFERENCES: RDPROG

FILES: NSQNTL, NOUT, NRAN, NRSI, NRSO, NRTI

ROUTINE NAME: READHD

DESCRIPTION: This routine reads the header records on the restart tapes.

CALLING SEQUENCE: CALL READHD (NUNIT, LABEL, ICALC)

NUNIT - Limit to be read (NSRI, NRTI)  
LABEL - Type of data record (FF, DI, RB,  
CM, GB, SF).  
ICALC - Resume calculations flag

REFERENCES: All restartable segments .

FILES: NOUT

ROUTINE NAME: RETRVE

DESCRIPTION: This routine retrieves node number, area and surface property arrays from the NRARR file and places it in core.

CALLING SEQUENCE: CALL RETRVE (IR)

IR - Flag indicating arrays are combined or  
uncombined

REFERENCES: Node combining routines

FILES: NRARR, NOUT

ROUTINE NAME: RIOPAC

DESCRIPTION: This routine was written for the Univac computer to simulate the CDC random access package. Contains entry points WRITHM and READMS.

CALLING SEQUENCE: CALL RIOPAC (NUNIT, FWA, NWDS, NR)

NUNIT - Unit number  
FWA - Address of first word to read  
NWDS - Number of words to be read  
NR - Record number

ROUTINE NAME: RKDATA

DESCRIPTION: This routine is user-called in the Operations Data block and defines parameters for the RKCAL link.

CALLING SEQUENCE: CALL RKDATA (NSGBIR, PNCH, FMIN, IRKN, RKSPC, NSPAC, SIG, AMPF, TAPE)

KEY VARIABLES: See Users Manual, Appendix D

REFERENCES: Operations Data Block

ROUTINE NAME: RSRCCK

DESCRIPTION: This routine determines if a parity error was encountered or a valid record was just read on a restart file.

CALLING SEQUENCE: CALL RSRCCK (IFILE, IOK)

IFILE - File name (NRSI, NRTI)  
IOK - Error/no error flag

REFERENCES: All restart tape read routines.

FILES: NRSI, NRTI, NOUT

ROUTINE NAME: RSTON

DESCRIPTION: This entry point in subroutine RESTRT sets up flags to resume reading the master restart tape, at the user's option.

CALLING SEQUENCE: CALL RSTON (See Users Manual, Appendix D).

REFERENCES: ODPFROG (Users Operations Data)

ROUTINE NAME: RSTOFF

DESCRIPTION: This entry point into subroutine RESTRT sets up flags to interrupt the reading of the master restart file at the user's direction.

CALLING SEQUENCE: CALL RSTOFF (reference Users Manual, Appendix D).

REFERENCES: ODPROG (Users Operations Data)

ROUTINE NAME: SATURD

DESCRIPTION: This routine sets up the planet parameters concerning the planet Saturn.

CALLING SEQUENCE: CALL SATURD

KEY VARIABLES: PRAD - Planet radius  
SOL - Solar constant  
PALB - Planet albedo factor  
WDS - Planet darkside emissive power  
WSS - Planet subsolar emissive power  
GRAV - Gravitational constant

REFERENCES: ORBIT1, ORBIT2, DITTP

ROUTINE NAME: SETBEA

DESCRIPTION: This routine positions a plot beam at the coordinates of a point in the subject mapping.

CALLING SEQUENCE: CALL SETBEA (X, Y)

X, Y - Coordinates of point

REFERENCES: NPAXES, NPFPLT, NPSCAL, OPAXES, OPFPLT, OPSCAL,  
OPVCS, PLDRIV

ROUTINE NAME: SETFLG

DESCRIPTION: This subroutine decodes the NMODIR (Z, ID) word to obtain FFCMB and IAUTO C flags as needed for appropriate node property access.

CALLING SEQUENCE: CALL SETFLG

REFERENCES: All segments

ROUTINE NAME: SFELAV

DESCRIPTION: This routine computes the position and area vectors for elements on a node given the nodal dimensions.

CALLING SEQUENCE: CALL SFELAV (ILP, DATA, TRAN, RX, RY, RZ, NB, NG)

ILP	- Surface type
DATA	- Array of surface dimensions
TRAN	- Direct cosine matrix for node
RX, RY, RZ	- Position vector for node
NB, NG	- Number of elements in the two directions the node is to be divided into

REFERENCES: SFMAIN

ROUTINE NAME: SFELEM

DESCRIPTION: This routine, given the nodal dimensions and the total number of elements required, determines the number of elements in each direction.

CALLING SEQUENCE: CALL SFELEM (ILP, DATA, TRAN, RX, RY, RZ, NTOT)

ILP	- Node type
DATA	- Array of node dimensions
TRAN	- Direction cosine matrix for node
RX, RY, RZ	- Position vector for node
NTOT	- Total number of elements required on node

REFERENCES: SFMAIN

ROUTINE NAME: SFELMT

DESCRIPTION: This routine establishes the minimum number of elements on a node to determine if a view is possible.

CALLING SEQUENCE: CALL SFELMT (NB, NG)

NB, NG - Minimum number of elements required to provide a representative view from the node

REFERENCES: SFMAIN

ROUTINE NAME: SFELSL

DESCRIPTION: This routine, given the total number of elements and the nodal dimensions, computes the number of elements in each direction to provide near-square elements.

CALLING SEQUENCE: CALL SFELSL (NB, NG, ILP, DATA, NTOT)

NB, NG - Computed number of elements required in various directions

ILP - Surface type

DATA - Array of surface dimensions

NTOT - Total number of elements to be used on the node

REFERENCES: SFELEM

ROUTINE NAME: SFGTST

DESCRIPTION: This integer function routine, given the allowable range on gamma concerning shadowing and given a gamma value, returns a 0 if the gamma value falls within the allowable range or a 1 if the value is outside the allowable range.

CALLING SEQUENCE: SFGTST (GN, GX, GT)

GN, GX - Minimum, maximum range

GT - Value to be checked

REFERENCES: SFSHAD

ROUTINE NAME: SFMAIN

DESCRIPTION: This is the main driving routine in the SFCAL link and directs the logic flow necessary to generate a shadow tape.

CALLING SEQUENCE: CALL SFMAIN

REFERENCES: SFPROG (preprocessor-generated)

ROUTINE NAME: SFPAKA

DESCRIPTION: This routine packs 9 solar waveband shadow-factor values into one word and generates 19 words each time the routine is called. These 19 words are then written to NPLS and NRSO (if requested) once every 10 calls. Entry point SFFLSH is called to write the last record and complete the writing sequence.

CALLING SEQUENCE: CALL SFPAKA (TABSHA, ICNT)

TABSHA - Unpacked shadow factor array  
ICNT - A counter that indicates when shadow data for 10 nodes have been packed

REFERENCES: SFMAIN

ROUTINE NAME: SFPAKE

DESCRIPTION: This routine is analogous to SFPAKA for the infrared waveband.

CALLING SEQUENCE: CALL SFPAKE (TABSHI, ICNT)

REFERENCES: SFMAIN



ROUTINE NAME: SFPL

DESCRIPTION: This routine computes the shadow factor to a planet element when interpolation of the shadow tables is doubtful.

CALLING SEQUENCE: CALL SFPL (ASFE, ASFA, IEP, KBS)

ASFE - Infrared waveband shadow factor  
ASFA - Solar waveband shadow factor  
IEP - Planet element pointer  
KBS - Node can be shaded flag

REFERENCES: DICALP

ROUTINE NAME: SFPROG

DESCRIPTION: This routine is generated by the preprocessor into the SFCAL link, and provides the necessary labeled commons.

CALLING SEQUENCE: CALL SFPROG

REFERENCES: TRASYS (root segment)

ROUTINE NAME: SFPRTR

DESCRIPTION: This routine prints out shadow factor tables that are read in from the master restart tape.

CALLING SEQUENCE: CALL SFPRTR

REFERENCES: SFMAIN

FILES: NSCR1, NOUT

ROUTINE NAME: SEPSHS

DESCRIPTION: This routine determines which surfaces could possibly shadow between node i and the sun.

CALLING SEQUENCE: CALL SFPSHS (RADS, POS, SUNP, NSURF, NSHAD, IN)

RADS - Radius of sphere encompassing node i  
POS - Position vector of encompassing sphere  
SUNP - Position vector for sun  
NSURF - Number of shadowing surfaces  
NSHAD - Number of possible shadowers  
IN - Sequence number of node i

KEY VARIABLES: ISHAD - Array of possible shadowers

REFERENCES: SFMAIN

ROUTINE NAME: SFRDIN

DESCRIPTION: This routine checks for restart information and sets the proper flags for the program. The shadowing surfaces are also set up in terms of the labeled commons.

CALLING SEQUENCE: CALL SFRDIN

REFERENCES: SFMAIN

ROUTINE NAME: SFRDRQ

DESCRIPTION: This routine reads in the restart data for one node at a time.

CALLING SEQUENCE: CALL SFRDRQ (IFLG, TABSHA, IN)

IFLG - Data check flag  
= 2HNO No data found  
= 3HYES Data found  
TABSHA - Array of restart data for node i  
IN - Sequence number for node i

REFERENCES: SFMAIN

ROUTINE NAME: SFSHAD

DESCRIPTION: This routine computes element-to-element shadowing between node i and the sun.

CALLING SEQUENCE: CALL SFSHAD (RX, RY, RZ, WE, WA, IN, I, NSS, RS)

RX, RY, RZ - Components of vector from element on  
node i to sun  
WE, WA - Elemental shadowing factors for IR and  
solar  
IN - Surface sequence number  
I - Element sequence number  
NSS - Number of possible shadowers  
RS - Square of the magnitude of the element-  
to-sun connecting vector

REFERENCES: SFMAIN

ROUTINE NAME: SFTRS3

DESCRIPTION: This routine transforms points in an SCS to points in the ICS, BCS, or CCS.

CALLING SEQUENCE: CALL SFTRS3 (X, Y, Z, A, B, C, RX, RY, RZ, TRAN)

X, Y, Z - Coordinates of a point in the ICS, BCS,  
or CCS  
A, B, C - Coordinates of the point in the SCS  
RX, RY, RZ - Components of the SCS origin position  
vector in the ICS, BCS, or CCS  
TRAN - Matrix of direction cosines

REFERENCES: SFELAV, SFELEM

ROUTINE NAME: SFUNCT

DESCRIPTION: This function, given a clock angle and a cone angle, interpolates the packed clock angle-cone angle array located in labeled common, computes the function value, A and stores the value in SFT.

CALLING SEQUENCE: A = SFUNCT (CL, CO, SFT)

CL - Clock angle  
CO - Cone angle  
SFT, A - Interpolated value

REFERENCES: DICALS, DICALP

ROUTINE NAME: SHADPT

DESCRIPTION: This routine computes the shadow points, given the orbit definition.

CALLING SEQUENCE: CALL SHADPT

REFERENCES: ORBIT1, ORBIT2

ROUTINE NAME: SINX

DESCRIPTION: This routine computes the sine of an argument, in degrees.

CALLING SEQUENCE: CALL SINX (A)

A - Argument, in degrees

ROUTINE NAME: SKFILE

DESCRIPTION: This routine advances a unit N number of files.

CALLING SEQUENCE: CALL SKFILE (NUNIT, N)

NUNIT - Unit number  
N - Number of files

REFERENCES: FINDST, STORE, QOMAIN, DITTP, PLLOAD

ROUTINE NAME: SORTDL

DESCRIPTION: This routine numerically sorts a doublet array within itself.

CALLING SEQUENCE: CALL SORTDL (IA, NA)

IA - Doublet array  
NA - Total number of elements

REFERENCES: QOMAIN, PLLOAD, RCPSS1

ROUTINE NAME: SORTS

DESCRIPTION: This routine numerically sorts a singlet array.

CALLING SEQUENCE: CALL SORTS (A, JJ)

A - Singlet input array  
JJ - Number of elements in the array

REFERENCE: NDUPCK

ROUTINE NAME: SPIN

DESCRIPTION: This routine is user-callable and allows the user to define the spin axis, spin rate, and start time.

CALLING SEQUENCE: CALL SPIN (CLOC, CON, RAT, ANGLE, TIMS)

CLOC - Clock angle locating spin axes  
CON - Cone angle locating spin axes  
RAT - Spin rate  
ANGLE - True anomaly angle where spin begins  
TIMS - Start time of spin

REFERENCES: User's Operations Data block

ROUTINE NAME: STFAQ

DESCRIPTION: This routine is user-called in the Operations Data block to generate a duplicate orbit point, given a true anomaly/ time and a step number to retrieve the data from.

CALLING SEQUENCE: CALL STFAQ (ANGLE, TIM, NST)

ANGLE - True anomaly the data are to be stored under  
TIM - Current orbital time data are to be stored under  
NST - Step number to retrieve data from

REFERENCES: Operations Data block (user-called)

FILES: NDI, NOUT, NSCR1, NTQ

ROUTINE NAME: STFFLG

DESCRIPTION: This routine stuffs bits into NMODIR (Z, ID) for the FFCMB and IAUTO C flags.

CALLING SEQUENCE: CALL STFFLG (NBITS)

NBITS - Bit location of FFCMB and IAUTO C flags in NMODIR (Z, ID) word.

REFERENCES: All segments.

ROUTINE NAME: STORE

DESCRIPTION: This routine generates the header record on the required file. If the file has been rewound or read, STORE repositions the file to the proper point.

CALLING SEQUENCE: CALL STORE (NUNIT, ISTEP, LABEL1, LABEL2)

NUNIT - Unit to write header on

ISTEP - Step number

LABEL1 - Identifier

LABEL2 - Identifier

REFERENCES: STFAQ, FFRDIN, SFRDIN, DIEND, DIRDRQ, GBSCFA, AQMAIN, ADSURF, CMCMBN, CMRSI, GBAPRX, GBR SI, RBRDIN, RBR SI

ROUTINE NAME: SUND

DESCRIPTION: This routine defines the parameters concerning the sun.

CALLING SEQUENCE: CALL SUND

KEY VARIABLES: PRAD - Planet radius

RSUN - Sun radius

WSUN - Sun emissive power

PALB - Sun albedo factor (0.0)

REFERENCES: ORBIT1, ORBIT2, DITTP

ROUTINE NAME: SURFP

DESCRIPTION: This routine sets up the spin rate, spin axis, parameters and computes dawn and dusk times for the planet surface option.

CALLING SEQUENCE: CALL SURFP (PNAM, AL, SUNL, AT)

PNAM - Planet name  
AL - Latitude on surface  
SUNL - Sun latitude  
AT - Atmospheric extinction coefficient

REFERENCES: User-called

FILES: NOUT

ROUTINE NAME: SYMBOL

DESCRIPTION: This routine generates Hollerith data on the plot frame. The sequence of characters is terminated by a (\$.).

CALLING SEQUENCE: CALL SYMBOL (A)

A - Array of Hollerith data

REFERENCES: NPMAIN, NPAXES, NPINFO, OPMAIN, OPAXES, OPINFO, OPVCS, PLGRID, PLSYMB

ROUTINE NAME: TPERR

DESCRIPTION: This routine writes out a message when an error is encountered reading a data file.

CALLING SEQUENCE: CALL TPERR (NUNIT, ISUB)

NUNIT - Data file where error was encountered  
ISUB - Name of subroutine attempting to read file

REFERENCES: All segments.

FILES: NOUT

ROUTINE NAME: TRANSF

DESCRIPTION: This routine combines the direction cosine matrices to generate a final C-S and S-C matrix.

CALLING SEQUENCE: CALL TRANSF (RMASS, TRANCB, BX, BY, BZ, TRANCS, TRANBS)

RMASS - Array of surface data  
TRANCB - Direction cosine matrix central to block system  
BX, BY, BZ - Block coordinate component vector  
TRANCS - Direction cosine matrix central to surface  
TRANBS - Direction cosine matrix block to surface

REFERENCES: BUILDG

ROUTINE NAME: TRASYS

DESCRIPTION: This is the root segment defined by the preprocessor enabling the operations data to direct all logic flow.

CALLING SEQUENCE: CALL TRASYS



ROUTINE NAME: TRNSP

DESCRIPTION: This routine transposes matrix A and stores the result in matrix B.

CALLING SEQUENCE: CALL TRANSP (A, B)

A - Input matrix  
B -  $A^T$

REFERENCES: DILOC

ROUTINE NAME: URANUD

DESCRIPTION: This routine defines the parameters necessary to orbit Uranus.

CALLING SEQUENCE: CALL URANUD

KEY VARIABLES: PRAD - Planet radius  
SOL - Solar constant  
PALB - Planet albedo factor  
WDS - Darkside emissive power  
WSS - Subsolar emissive power  
GRAV - Gravitational constant

REFERENCES: ORBIT1, ORBIT2, DITTP

ROUTINE NAME: VENUSD

DESCRIPTION: This routine defines the parameters necessary to orbit Venus.

CALLING SEQUENCE: CALL VENUSD

KEY VARIABLES: PRAD - Planet radius  
SOL - Solar constant  
PALB - Albedo factor  
WDS - Darkside emissive Power  
WSS - Subsolar emissive power  
GRAV - Gravitational constant

REFERENCES: ORBIT1, ORBIT2, DITTP

ROUTINE NAME: WRAPUP

DESCRIPTION: This routine sets up the normal exit from the TRASYS processor and prints some information messages.

CALLING SEQUENCE: CALL WRAPUP

REFERENCES: TRASYS (root segment)

FILES: NOUT

ROUTINE NAME: WRITHD

DESCRIPTION: This routine writes header records to the restart tapes.

CALLING SEQUENCE: CALL WRITHD (NUNIT, LABEL)

NUNIT - Name of file to write header record on

LABEL - Type of information to follow header

REFERENCES: All restart segments.

ROUTINE NAME: WRITHT

DESCRIPTION: This routine writes trailer records after information on the restart files.

CALLING SEQUENCE: CALL WRITHT (NUNIT, LABEL)

REFERENCES: All restart segments.

ROUTINE NAME: ZNPMAX

DESCRIPTION: This function routine is called from the node plotter (NPLOT) and scales the nodes to be plotted.

CALLING SEQUENCE: ZNPMAX (NNP)

NNP - Number of surface/nodes to be selectively plotted

REFERENCES: NPMAIN, NPCONV

C. FILE DEFINITIONS - PROCESSOR LIBRARY

FILE NAME: BCDOL

PROGRAM VARIABLE NAME: NBCDOU

UNIT REFERENCE (UNIVAC/JSC): 29

PURPOSE: This file is the output tape for BCD data in the thermal analyzer input format.

SEGMENT REFERENCES: QOCAL (WRITE)  
RKCAL (WRITE)  
RCCAL (WRITE)

FILE NAME: DI

PROGRAM VARIABLE NAME: NDI

UNIT REFERENCE (UNIVAC/JSC): 10

PURPOSE: This file is for storage of solar, planetary, and albedo direct irradiation data.

SEGMENT REFERENCES: DICAL (WRITE)  
DRCAL (WRITE)  
AQCAL (READ)  
PLOT (READ)

FILE NAME: DIR

PROGRAM VARIABLE NAME: NDIR

UNIT REFERENCE (UNIVAC/JSC): 22

PURPOSE: This file is for storage of direct irradiation data input through the flux data block.

SEGMENT REFERENCES: DICAL (READ)

FILE NAME: FF

PROGRAM VARIABLE NAME: NFF

UNIT REFERENCE (UNIVAC/JSC): 9

PURPOSE: This file is used to store form factor data.

SEGMENT REFERENCES: FFCAL (WRITE)  
GBCAL (READ)  
RBCAL (WRITE)

FILE NAME: FFR

PROGRAM VARIABLE NAME: NFFR

UNIT REFERENCE (UNIVAC/JSC): 21

PURPOSE: This file is used to store form factor data input through  
the form factor data blocks.

SEGMENT REFERENCES: FFCAL (READ)

FILE NAME: GBIR

PROGRAM VARIABLE NAME: NGBIR

UNIT REFERENCE (UNIVAC/JSC): 11

PURPOSE: This file is for storage of infrared waveband gray-body  
factor data.

SEGMENT REFERENCES: AQCAL (READ)  
GBCAL (WRITE)  
ODPROG - Subroutine GBAPRX (WRITE)  
RCCAL (READ)  
RKCAL (READ)

FILE NAME: GBIRR

PROGRAM VARIABLE NAME: NGBIRR

UNIT REFERENCE (UNIVAC/JSC): 23

PURPOSE: This is the correspondence data storage file.

SEGMENT REFERENCES: RCCAL (READ)  
RKCAL (READ)  
QOCAL (READ)

FILE NAME: GBSO

PROGRAM VARIABLE NAME: NGBSO

UNIT REFERENCE (UNIVAC/JSC): 12

PURPOSE: This file is for storage of solar-waveband gray-body factor data.

SEGMENT REFERENCES: AQCAL (READ)  
GBCAL (WRITE)  
ODPROG - Subroutine GBAPRX (WRITE)

FILE NAME: OUTPUT

PROGRAM VARIABLE NAME: NOUT

UNIT REFERENCE (UNIVAC/JSC): 6

PURPOSE: This is the print output file.

SEGMENT REFERENCES: A11

FILE NAME: PLS

PROGRAM VARIABLE NAME: NPLS

UNIT REFERENCE (UNIVAC/JSC): 13

PURPOSE: This is used to store the spacecraft/planet form factor matrix and shadow factor data.

SEGMENT REFERENCES: DICAL (READ/WRITE)  
SFCAL (READ/WRITE)

FILE NAME: PLSR

PROGRAM VARIABLE NAME: NPLSR

UNIT REFERENCE (UNIVAC/JSC): 25

PURPOSE: This file is used to store data from the shadow factor data block.

SEGMENT REFERENCES: SFCAL (READ)

FILE NAME: PUNCH

PROGRAM VARIABLE NAME: NPUN

UNIT REFERENCE (UNIVAC/JSC): 7

PURPOSE: This file is the punch output file.

SEGMENT REFERENCES: DICAL (WRITE)  
FFCAL (WRITE)  
RCCAL (WRITE)  
RKCAL (WRITE)  
QOCAL (WRITE)

FILE NAME: RIO

PROGRAM VARIABLE NAME: NRAN

UNIT REFERENCE (UNIVAC/JSC): 8

PURPOSE: This file is the primary random access file, which is used to store all node and surface description data.

SEGMENT REFERENCES: DICAL (READ)            NPLOT (READ)  
                     DRCAL (READ)          OPLOT (READ)  
                     FFCAL (READ)          RBCAL (READ)

FILE NAME: RSI

PROGRAM VARIABLE NAME: N 3I

UNIT REFERENCE (UNIVAC/JSC): 14

PURPOSE: This file is the master restart input tape.

SEGMENT REFERENCES: DICAL (READ), FFCAL (READ), GBCAL (READ),  
                     SFCAL (READ), CMCAL (READ), RBCAL (READ),  
                     DRCAL (READ)

FILE NAME: RSO

PROGRAM VARIABLE NAME: NRSO

UNIT REFERENCE (UNIVAC/JSC): 15

PURPOSE: This file serves as the restart output tape.

SEGMENT REFERENCES: DICAL (WRITE), FFCAL (WRITE), GBCAL (WRITE),  
                     SFCAL (WRITE), RBCAL (WRITE), DRCAL (WRITE)

FILE NAME: RTI

PROGRAM VARIABLE NAME: NRTI

UNIT REFERENCE (UNIVAC/JSC): 17

PURPOSE: This file is the temporary restart input tape.

SEGMENT REFERENCES: DICAL (READ), FFCAL (READ), GBCAL (READ)  
RBCAL (READ), DRGAL (READ)

FILE NAME: RTO

PROGRAM VARIABLE NAME: NRTO

UNIT REFERENCE (UNIVAC/JSC): 18

PURPOSE: This file is the temporary restart output tape.

SEGMENT REFERENCES: DICAL (WRITE), FFCAL (WRITE), GBCAL (WRITE)



FILE NAME: SONTL

PROGRAM VARIABLE NAME: NSQNTL

UNIT REFERENCE (UNIVAC/JSC): 16

PURPOSE: This file contains pointers for the random access file, plus miscellaneous flags and quantities generated in the preprocessor for use by processor segments.

SEGMENT REFERENCES: RDPROG (READ)

FILE NAME: TAPE1

PROGRAM VARIABLE NAME: NSCR1

UNIT REFERENCE (UNIVAC/JSC): 1

PURPOSE: This file is scratch file 1. Scratch files are never used to pass information between segments.

SEGMENT REFERENCES: GBCAL (READ/WRITE)  
RCCAL (READ/WRITE)  
RKCAL (READ/WRITE)

FILE NAME: TAPE2

PROGRAM VARIABLE NAME: NSCR2

UNIT REFERENCE (UNIVAC/JSC): 2

PURPOSE: This file is scratch file 2.

SEGMENT REFERENCES: GBCAL (READ/WRITE)  
PLOT (READ/WRITE)  
QOCAL (READ/WRITE)

FILE NAME: TAPE3

PROGRAM VARIABLE NAME: NSCR3

UNIT REFERENCE (UNIVAC/JSC): 3

PURPOSE: This file is scratch file 3.

SEGMENT REFERENCES: FFCAL (READ/WRITE)  
GBCAL (READ/WRITE)  
RCCAL (READ/WRITE)

FILE NAME: TQ

PROGRAM VARIABLE NAME: NTQ

UNIT REFERENCE (UNIVAC/JSC): 14

PURPOSE: This file is used for storage of absorbed heat data.

SEGMENT REFERENCES: AQCAL (WRITE)  
QOCAL (READ)  
PLOT (READ)

FILE NAME: TQR

PROGRAM VARIABLE NAME: NTQR

UNIT REFERENCE (UNIVAC/JSC): 26

PURPOSE: This file is a restart file for absorbed heat data.

SEGMENT REFERENCES: Not currently used. Name and unit reserved

FILE NAME: TRAJ

PROGRAM VARIABLE NAME: NTRAJ

UNIT REFERENCE (UNIVAC/JSC): 4

PURPOSE: This file is used to input trajectory tape data.

SEGMENT REFERENCES: ODPROG - Subroutine DITTP (READ)

FILE NAME: USER1

PROGRAM VARIABLE NAME: NUSER1

UNIT REFERENCE (UNIVAC/JSC): 19

PURPOSE: This is a scratch file reserved for the user.

SEGMENT REFERENCES: User option

FILE NAME: USER2

PROGRAM VARIABLE NAME: NUSER2

UNIT REFERENCE (UNIVAC/JSC): 20

PURPOSE: This is a scratch file reserved for the user.

SEGMENT REFERENCES: User option

D. VARIABLE DEFINITIONS - PROCESSOR LIBRARY

LABELED COMMON /ALPH/

This common block contains an array of nodal absorptivities in the solar waveband.

ALPH - An array of solar absorptivities for active nodes

LABELED COMMON /AQQDP/

This common block provides a storage area in the AQPROG segment for incident planetary fluxes that are read in from the NDI file.

QDP - An array of incident planetary fluxes

LABELED COMMON /AQQDR/

This common block provides a storage area in the AQPROG segment for incident albedo fluxes that are read in from the NDI file.

QDR - An array of incident albedo fluxes

LABELED COMMON /AQQDS/

This common block provides a storage area in the AQPROG segment for incident solar fluxes that are read in from the NDI file.

QDS - An array of incident solar fluxes

LABELED COMMON /AQTEMP/

This common block provides temporary storage in the AQPROG segment for the node array, as read from the NDI file, for verifying correspondence data with the active model node array, NODE.

ITEMP - Temporary array of node numbers

#### LABELED COMMON /AREA/

This common block contains an array of the active model nodal areas.

AREA - An array of nodal areas

#### LABELED COMMON /AREAT/

This common block provides a temporary working and storage area for use in the QOPROG segment when combining nodal areas.

AREAT - An array of combined nodal areas

#### LABELED COMMON /ARRAYS/

This common block is set up by the preprocessor and contains user-input arrays from the Array Data block.

ADUMMY     - Integer count of the first array  
NAME1       - First user-input array  
NAME1 (N)   - Integer count of the second array  
NAME2       - Second user-input array

#### LABELED COMMON /BCSN/

This common block contains all the user-input block coordinate system (BCS) names and/or the default BCS name, ALLBLK.

#### LABELED COMMON /BLOCK/

This common block contains blocking information used by blank common to invert a matrix when calculating gray-body factors.

- NBLCK - Maximum number of blocks into which a matrix may be divided (set by a data statement in GBMAIN)
- NBUP - Minimum number of blocks in an upper triangular matrix, as determined by full utilization of blank common
- NBLO - Minimum number of blocks in a lower triangular matrix, as determined by full utilization of blank common

#### LABELED COMMON /CAL280/

This common block contains plotting information.

- XC - The X-coordinate of the point to be plotted in the plotter coordinate system
- YC - The Y-coordinate of the point to be plotted in the plotter coordinate system
- IP - The index on a computed GO TO statement used in NPFPLT and OPFPLT

#### LABELED COMMON /BFA/

This common block contains form factor blockage (shadow) factors in the solar waveband.

#### LABELED COMMON /BFE/

This common block contains form factor blockage (shadow) factors in the infrared waveband.

#### LABELED COMMON /BLKDIR/

This common block contains the block coordinate system directory.

- NBLKDR (1, I) - Block coordinate system name
- NBLKDR (2, I) - Length of the block data written on the random access record
- NBLKDR (3, I) - Random access record number
- NBLKDR (4, I) - Flag to indicate whether or not the BCS transformation data have been applied

## LABELED COMMON / CCONST/

This common block contains a collection of constants and variables used throughout the processor.

AQPRNT	- Print/no print flag for absorbed flux data printout
ARAD	- Not used
DELCT	- Not used
DIACC	- Element selection accuracy factor for node-to-planet form factors
DIACCS	- Element selection accuracy factor for direct flux shadowing calculations
DIPNOSH	- Shadow/no shadow flag for direct flux calculations
DIPNCH	- Direct flux punch flag
DLTLINE	- Not used
DTR	- Conversion factor for degrees to radians
ELPBEA	- Not used
FFACC	- Element selection accuracy factor for node-to-node form factor calculations
FFACCS	- Element selection accuracy factor for form-factor shadowing calculations
FFCMB	- Form factor combined/not combined flag
FFDISF	- Flag to indicate whether FF, DI or SF segments have been executed
FFMIN	- Minimum form factor value to be saved. All form factors smaller than FFMIN are set to zero
FFNAC	- Flag to allow the input of form factors from an RSI file with a node array that does not match the node array that resides in core
FFNOSH	- Shadow/no shadow flag for form-factor calculations
FFPNCH	- Form-factor punch flag
FFPRNT	- Form-factor print flag
FFRATL	- Maximum allowable ratio of the maximum number of elements indicated on a node pair divided by the number of elements indicated by arithmetic averaging. If this ratio exceeds FFRATL, the two nodes are temporarily subdivided
FFZERO	- The initialization value for the form factor request matrices (defaults to -1.0)
FOG	- Not used
GAUSS	- Not used
GBWBND	Waveband definition for gray-body calculations
IAI	- Step number indicating where the IR gray-body factor matrix is to be obtained for flux calculations using the ORBGEN option
IALBFL	- Albedo flux compute/stuff flag
IAQGBI	- Step number from which IR gray-body factors are to be obtained from absorbed-Q calculations
IAQGBS	- Step number from which solar gray-body factors are to be obtained for absorbed-Q calculations

IAQSDA	- Step number from which direct albedo fluxes are to be obtained for absorbed-Q calculations
IAQSDP	- Step number from which direct planetary fluxes are to be obtained for absorbed-Q calculations
IAQSDS	- Step number from which direct solar fluxes are to be obtained for absorbed-Q calculations
IAS	- Step number indicating where the solar gray-body factor matrix is to be obtained for flux calculations using the ORBGEN option
IAUTOC	- Flag indicating whether to include autocombining (polygon combining) in form factor combining operations
ICMBL	- <del>Number of combinations defined in correspondence data</del>
IEQFF	- Flag pointing to equivalent form factors
IFFSHO	- Flag indicating whether or not to compute form factors to shadower-only nodes. Also used to specify whether or not shadower-only nodes are to be plotted.
IGBSFF	- Step number from which form factors are to be obtained for use in gray-body calculations
ILLUMN	- Not used
IMESS	- Starting address of the secondary MESS node array
INCORE	- Flag indicating whether combined or uncombined node, area, and properties arrays are in core
INTMF	- Not used
IOVL	- Index on a computed GO TO statement set in ODPROG and used in TRASYS. This determines which segment is to be called
IPRDMP	- Flag used to specify the waiting of node, area, and properties arrays on the RSO file following calls to BUILDG, ADD, or MODXX routines
IQOARY	- Array of step numbers where absorbed-Q data are stored
IQOCOR	- Step number from which correspondence data are to be obtained
IQOTAR	- Initial array number for the output Qs (= IQOTME + 1)
IQOTME	- Output time array number
IPLAFL	- Planetary flux compute/stuff flag
IRKCN	- Initial radiation conductor number
IRKNGB	- Step number from which gray-body factors are to be obtained for use in radiation conductor calculations
IRKNSP	- Space node number
IRSI	- RSI file designator
IRTI	- RTI file designator
ISFAC	- Flag to specify whether or not direct flux shadow factors are to be written on the RSO file
ISKIP	- Flag used to skip data on the RSO file until the proper header file is found
ISOLFL	- Solar flux compute/stuff flag
ISPND	- Starting address of the primary MESS node array
ISTR	- Not used
ITRALL	- Not used



ITRCAO	- Trace flag for the AWPORG segment
ITRCBO	- Trace flag for the QOPORG segment
ITRCCO	- Trace flag for the RBPROG segment (not used)
ITRCDO	- Trace flag for the PLPROG segment
ITRC10	- Trace flag for routines FINDST and STORE
ITRC20	- Trace flag for BUILDG
ITRC30	- Trace flag for the FFPROG segment (not used)
ITRC40	- Trace flag for the SFPROG segment (not used)
ITRC50	- Trace flag for the NPPROG and RCPORG segments
ITRC60	- Trace flag used to print the original radiation conductors in the RCPORG segment
ITRC70	- Trace flag for the DIPORG setment
ITRC80	- Trace flag for the gray-body calculations
ITRC90	- Trace flag for the RKPROG segment
MAXBC	- Length of blank common
MB	- <del>Not used</del>
MFCO	- Not used
MFLUX	- Not used
MG	- Not used
MIT SIN	- Flag to specify MITAS or SINDA output formats
MNND	- Total number of nodes defined in the surface data block
MRSP	- Not used
MSRF	- Not used
NBCDSK	- Not used
NBLKLN	- Number of block coordinate systems in the model
NCONST	- Not used
NELN	- Not used
NERN	- Effective radiation node number
NFFTYF	- Type flag for form factors (FF, RB, CM)
NFIGCO	- Configuration name for correspondence data access
NFIGFF	- Configuration name for form factor access
NFIGGB	- Configuration name for gray-body factor access
NIBBLE	- Not used
NMESS	- Number of MESS node pairs
NMIR	- Not used
NMODLS	- Number of configurations created by BUILDG/ADD sequences (maximum of 20 allowed)
NN	- Total number of nodes defined by BUILDG/ADD
NNOD	- Number of active nodes in the model
NNODC	- Number of nodes after combining
NNODU	- Number of nodes before combining
NRMOD	- Flag defining the modification level of an RSO/RSI restart file
NRSP	- Not used
NS	- Total number of surfaces defined by BUILDG/ADD
NSFO	- Configuration name to be used to identify the file when writing the shadow-factor output tape (SHADO)
NSFT	- Configuration name to identify the desired file on the shadow-factor input tape (SHADI)

NSPND - Number of MESS node pairs plus the number of special nodes  
 NSSTEP - Sequence number of the current substep  
 NSTEP - Sequence number of the current step  
 NSTPDI - Step number reference for DI data in DRCAL  
 NSTPL - Step number indicating where planetary fluxes are to be  
           obtained for a planet-oriented case using the ORBGEN option  
 NSTSOL - Step number indicating where solar fluxes are to be  
           o obtained for a sun-oriented case using the ORBGEN option  
 NSURF - Number of active shadowing surfaces in the model  
 PI - The constant  
 QOAMPF - Area multiplying factor for the output Qs  
 OOFMPF - Energy multiplying factor for the output Qs  
 QOPNCH - Flag to punch the output Qs  
 QORMPF - Not used  
 QOTAPE - Flag to write the output Qs on a BCD tape  
 QOTMPF - Time multiplying factor  
 QOTYPE - Flag to output the Q tables, the orbital average Q, or both  
 RALB - Multiplying factor for the absorbed albedo heat  
 RFRAC - Significant radiation fraction used in the RCPROG segment  
 RKAMPF - Area multiplying factor used in the RKPROG and RCPROG  
           segments  
 RKMIN - Minimum value of     /     that will result in a valid radia-  
           tion conductor  
 RKPCH - Radiation conductor punch flag  
 RKSP - Flag for calculating radiation conductors to space  
 RKTape - Flag to write radiation conductors to the BCD tape  
 RPLAN - Multiplying factor for the absorbed planetary heat  
 RSOLAR - Multiplying factor for the absorbed solar heat  
 RTD - Conversion factor for radians to degrees  
 RTOL - Tolerance flag for form factor combining (see Users Manual,  
           Appendix D, subroutine RCDATA)  
 SAOS - Not used  
 SFPRNT - Print/no print flag for shadow factors on restart tape  
 SIGMA - Stefan-Boltzman constant  
 STRACK - Not used  
 THGHT - Not used  
 TPIAM - Not used  
 TRUANF - True anomaly of the final point in the orbit as defined  
           using the ORBGEN option  
 TRUANI - True anomaly of the first point in the orbit as defined  
           using the ORBGEN option

#### LABELED COMMON /DIMS/

This common block contains an array of position vectors for shadowing surfaces.

DIMS - Position vector locating the origin of the shadowing-surface coordinate system in the central coordinate system

#### LABELED COMMON /DIRCT/

This common block contains the translation and rotation information relating a block coordinate system (BCS) to the central coordinate system (CCS), as well as a directory of random-access record numbers for the surfaces in the BCS.

DIRCT (1) - X-component of the translation vector  
DIRCT (2) - Y-component of the translation vector  
DIRCT (3) - Z-component of the translation vector  
DIRCT (4) - Rotation about the CCS X-axis  
DIRCT (5) - Rotation about the CCS Y-axis  
DIRCT (6) - Rotation about the CCS Z-axis  
DIRCT (7) -  
DIRCT (8) - } Order of rotation about the CCS X-, Y-, Z-axes,  
DIRCT (9) - } respectively  
DIRCT (10) - }  
          : - } Random-access record numbers  
          : - }  
          :  
          :

#### LABELED COMMON /DISUR1/

This common block is used to store surface data read in from the random-access file in the DIPROG segment

COMM1 - Five 6-character words describing the surface  
CSHDI - Can-shade flag  
DATAI - Array of the five surface description parameters,  
          ALPHA, BMIN, BMAX, GMIN, and GMAX  
DTEI - Date of the run  
DUMI (16) - Dummy array  
ILKI - Identifier for type of surface  
INOD - Can-be-shaded flag  
KSI - Sequence number of surface  
NRMSS1 (1) - Surface ID number  
OLDAI - Surface area  
POSNI - Position vector locating the center of the encompassing  
          sphere in the CCS

PROPI - Array of surface optical properties: solar absorptivity (ALPH), infrared emissivity (EMISS), infrared transmissivity (TIR), and solar transmissivity (TSO)  
 RADI - Radius of the preshadowing sphere encompassing the surface  
 RXI -  
 RYI - } Components of the position vector locating the origin of  
 RZI - } the SCS in the CCS  
 SPRII - Specular reflectance of the surface in the IR waveband  
 SPRSI - Specular reflectance of the surface in the solar waveband  
 TRANI - Transformation Matrix relating the SCS to the CCS

#### LABELED COMMON /DRQDP/

This common block provides a storage area in the DRCAL segment for incident planetary fluxes that are read in from the NDI file.

QDP - An array of incident planetary fluxes

#### LABELED COMMON /DRQDR/

This common block provides a storage area in the DRPROG segment for incident albedo fluxes that are read in from the NDI file.

QDR - An array of incident albedo fluxes

#### LABELED COMMON /DRQDS/

This common block provides a storage area in the DRPROG segment for incident solar fluxes that are read in from the NDI file.

QDS - An array of incident solar fluxes

#### LABELED COMMON /DRSHDC/

This common block is used to store the sequence numbers of possible shadowing surfaces in the preshadowing calculations of segment DRCAL.

ISHAD - An array of sequence numbers of possible shadowing surfaces

#### LABELED COMMON /DRSUN/

This common block contains information used in calculating the incident solar flux on a node in the DRPROG segment.

FACT - Unshadowed solar flux on a node  
Y - Square of the distance from the node to the sun  
Z - Ratio of the solar constant to the node-to-sun distance

#### LABELED COMMON /DRSURL/

This common block serves the same purpose in the DRPROG segment that DISURL serves in the DIPROG segment (see the DISURL description).

#### LABELED COMMON /DRTRAN/

This common block contains the matrix of direction cosines necessary to transform vectors defined in the planet-oriented vehicle coordinate system to vectors in the user-defined vehicle coordinate system.

PLDC - Transformation matrix

#### LABELED COMMON /DRTRSH/

This common block contains miscellaneous variables used to calculate incident fluxes in the DRPROG segment.

IN - Sequence number of the current node  
NCHECK - Not used  
NELT - Total number of elements used on the node  
NSHAD - Number of possible shadowing surfaces  
NSHADR - Not used  
SFAVT - Temporary array of elemental surface area vectors  
SFPVT - Temporary array of elemental surface position vectors  
SHADS - Shadow factor  
SUNPVT - Temporary sun position vector

#### LABELED COMMON /DRVCTR/

This common block contains vector information for use in calculating incident fluxes in the DRPROG segment.

NEPT - Not used  
NEST - Optimum number of elements used on the node  
SUNPV - Sun position vector  
SFAV - Array of elemental surface area vectors  
SFPV - Array of elemental surface position vectors

#### LABELED COMMON /DSTORE/

This common block contains information used in storing and retrieving data from units assigned to TRASYS.

IDSTR (I, 1) - Unit identifier (NUNIT)  
IDSTR (I, 2) - Number of files written to NUNIT  
IDSTR (I, 3) - Yes/no flag indicating whether or not the unit has been repositioned

#### LABELED COMMON /DSTR/

This common block contains surface description data for shadowing surfaces.

DSTR (1, I) - ALPHA  
DSTR (2, I) - BMIN  
DSTR (3, I) - BMAX  
DSTR (4, I) - GMIN  
DSTR (5, I) - GMAX

#### LABELED COMMON /EMISS/

This common block contains IR emissivities.

EMISS - Array of IR emissivities for the active nodes

#### LABELED COMMON /FA/

This common block contains one row of a symmetric matrix.

FA - Area-form factor product, or area-script F product

#### LABELED COMMON /FFDAT1/

This common block contains miscellaneous variables used in the FFCAL segment.

FANS - Unshadowed solar form factor  
FENS - Unshadowed IR form factor  
IABTIME - Abort flag set when maximum number of elements per node is exceeded  
ICALTP - Flag indicating whether the form factors came from cards, tape, were calculated, or were equivalenced  
IEOFFR - End-of-file flag on unit NFFR  
IN - Sequence number of the current "looker" node  
JN - Sequence number of the current "looker" node  
KTAE - Not used  
NEX - Flag to indicate when nodes have been divided into sub nodes  
TIMEF - Cp time at beginning of a form factor calculation  
TIMER - CP time at beginning of a row of form factor calculations  
TIMET - Time at the beginning of the form-factor calculations  
TPER - Not used

#### LABELED COMMON /FFEQ/

This common block contains the master index for the random-access file (NSCR3) used in equivalenced form factors.

INDXF - Master index

#### LABELED COMMON /FFSHDC/

This common block serves the same purpose in the FFPROG segment as DRSHDC serves in the DRPROG segment (see the DRSHDC description).

#### LABELED COMMON /FFSUMC/

This common block contains form-factor sums.

SUM - An array of form-factor sums

LABELED COMMON /FFVALI/

This common block contains a row of IR area-form factor products.

FFVALI - IR area-form factor products

LABELED COMMON /FFVALS/

This common block contains a row of solar area-form factor products.

FFVALS - Solar area-form factor products

LABELED COMMON /GBIR/

This common block contains a row of IR gray-body factors for use in the AQPROG segment.

GBIR - IR gray-body factors

LABELED COMMON /GBSO/

This common block contains a row of solar gray-body factors for use in the AQPROG segment.

GBSO - Solar gray-body factors

LABELED COMMON /IFS/

This common block contains an array of sequence numbers for shadowing surfaces.

IFS - Shadowing-surface sequence numbers

LABELED COMMON /IKS/

This common block contains an array of surface-type identifiers for shadowing surfaces.

IKS - Type of shadowing surface



LABELED COMMON /INDX/

This common block contains the master index for the random-access file NRAN.

INDX - Master index

LABELED COMMON /INDXN/

This common block contains an array of random-access record numbers for active nodes.

INDXN - Array of random-access record numbers for active nodes

LABELED COMMON /INDXS/

This common block contains an array of random-access record numbers for active surfaces.

INDXS - Array of random-access record numbers for active surfaces

LABELED COMMON /ISHAD/

This common block serves the same purpose in the DIPROG segment as DRSHDC serves in the DRPROG segment (see the DRSHDC description).

LABELED COMMON /ISPEC/

This common block contains an array of active specular-surface sequence numbers.

ISPEC - Array of sequence numbers of active specular surfaces

LABELED COMMON /ISPN/

This common block contains an array of primary MESS nodes and/or special nodes for use in the RCPROG segment.

ISPN - Array of primary MESS nodes and/or special nodes

LABELED COMMON /ISTPDR/

This common block contains a directory of user-assigned step numbers.

ISTPDR - Array of user-assigned step numbers

LABELED COMMON /JSURF/

This common block contains an array of all surface numbers defined by BUILDG/ADD.

JSURF - Array of surface numbers

LABELED COMMON /LNGSEG/

LAQSEG	- Length of AQ segment
LCMCOM	- Length of dynamic core for CM segment
LCMSEG	- Length of CM segment
LDISEG	- Length of DI segment
LDRSEG	- Length of DR segment
LFFSEG	- Length of FF segment
LGBCOM	- Length of dynamic core for GB segment
LGBSEG	- Length of GB segment
LMFSEG	- Not used
LNPSEG	- Length of NP segment
LODSEG	- Length of OD segment
LOPSEG	- Length of OP segment
LPLCOM	- Length of dynamic core for PL segment
LPLSEG	- Length of PL segment
LQOCOM	- Length of dynamic core for QO segment
LQOSEG	- Length of QO segment
LRBSEG	- Length of RB segment
LRCCOM	- Length of dynamic core for RC segment
LRCSEG	- Length of RC segment
LRDSEG	- Length of RD segment
LSFSEG	- Length of SF segment
MAXFL	- Maximum field length available
NCURFL	- Current field length

LABELED COMMON /MNP/

This common block contains a list of selected nodes to be plotted in the NPPROG segment.

MNP - Array of selected node numbers to be plotted

LABELED COMMON /MSND/

This common block contains an array of secondary MESS node numbers for use in the RCPRG segment.

MSND - Array of secondary MESS node numbers

LABELED COMMON /MSP/

This common block contains a list of selected surfaces to be plotted in the OPPROG segment.

MSP - Array of selected surface numbers to be plotted

#### LABELED COMMON /NCONST/

This common block contains miscellaneous variables for use in the NPPROG segment.

ITITLE - Title to be written on each plot frame  
KC - Flag indicating which view is to be plotted  
KERR - Scaling error flag  
LINPL - Line plot control flag  
NOMI - Values to indicate the order of rotation of the plotter  
NPHI - coordinate system through the angles OMI, PHI, and PSI,  
NPSI - respectively.  
OMI - Angle of rotation of the plotter coordinate system (PCS)  
about the X-axis  
PHI - Angle of rotation of the PCS about the Z-axis  
PSI - Angle of rotation of the PCS about the Y-axis  
SCL - Scale factor for plotting  
VROT - Transformation matrix of direction cosines resulting  
from the rotations PHI, PSI, and OMI. Transforms vectors  
in the PCS to the user-defined view

#### LABELED COMMON /NDS/

This common block is used in the RCPROG segment for temporarily storing node numbers.

NODET - Temporary node array

#### LABELED COMMON /NODE/

This common block contains an array of the active node numbers.

NODE - Array of active node numbers

#### LABELED COMMON /NOROLO/

This common block contains information pertinent to partitioning a lower triangular matrix for use in the GBPROG segment.

NLO (1, I) - Number of rows in block I  
NLO (2, I) - Number of elements in block I

#### LABELED COMMON /NOROU/

This common block contains information pertinent to partitioning an upper triangular matrix for use in the GBPROG segment.

NUP (1, I) - Number of rows in block I  
NUP (2, I) - Number of elements in block I

#### LABELED COMMON /NPMAS/

This common block is used to store surface data read in from the random-access file in the NPPROG segment.

CC - Transformation matrix relating the SCS to the CCS  
DATA - Array of the five surface description parameters, ALPHA, BMIN, BMAX, GMIN, and GMAX  
ILK - Identifier for type of surface  
RMAS - Dummy array of 11 words  
RX - }  
RY - } Components of the position vector locating the origin of  
RZ - } the SCS in the CCS

#### LABELED COMMON /NSPEC/

This common block contains the number of active specular surfaces.

NSPEC - Number of active specular surfaces

#### LABELED COMMON /OCONST/

This common block serves the same purpose in the OPPROG segment as NCONST serves in the NPPROG segment (see the NCONST description), except for SCL and the addition of two variables.

RPLN - Planet radius in plot frame dimensions  
SCL - Maximum spacecraft dimension, measured from the CCS origin, in plot frame dimensions  
SCLR - Orbit radius in plot frame dimensions

LABELED COMMON /ODTEMP/

This common block provides a scratch array available to the programmer within any given segment.

ODTEMP - Scratch array dimensioned to the maximum of 100 and the total number of nodes input in the Surface Data block

LABELED COMMON /OPMASS/

This common block serves the same purpose in the OPPROG segment as NPMASS serves in the NPPROG segment (see the NPMASS description)

LABELED COMMON /OPTRAN/

This common block contains variables used in the OPPROG segment.

S - Transformation matrix to locate the solar position vector in the plotter coordinate system

CTR - Transformation matrix to transform vectors in the CCS to vectors in the orbit coordinate system (OCS)

XV - }  
YV - } Components of the vehicle position vector in the OCS  
ZV - }

# LABELED COMMON /ORBIT/

This labeled common contains all variables associated with orbit definition and other orbit-associated parameters.

ALAN	- Longitude of the ascending node
APER	- Argument of perifocus
ASUN	- Not used
ATMT	- Atmospheric transmissivity
BETA	- Angle from the $Z_0$ -axis of the orbit coordinate system to the sun vector (vertex at planet center)
BETAS	- Same as BETA, except star vector
CIGMA	- Angle, in the orbital plane, from periapsis to the projection of the solar vector in the direction of the spacecraft's motion
CIGMAS	- Same as CIGMA, except star vector
CLOCK	- Clock angle to the spin vector
CONE	- Cone angle to the spin vector
DAWN	- Sunrise time
DUSK	- Sunset time
DWP	- Subsolar planet emissive power, less the darkside emissive power
ECC	- Orbit eccentricity
GRAV	- Planet gravitational constant
HA	- Altitude of apoapsis
HP	- Altitude of periapsis
ICALFL	- Not used
INSHAD	- Flag indicating whether the point is in or out of the planet shadow
IORBIT	- Flag for type of orbit = 1 Planet = 2 Sun = 3 Star = 4 Tape
IROTX	- Order of performing rotation ROTX (1, 2, or 3)
IROTY	- Order of performing rotation ROTY (1, 2, or 3)
IROTZ	- Order of performing rotation ROTZ (1, 2, or 3)
ISFT	- Flag directing the use of shadow factor data
ISKPSO	- Not used
NSPFF	- Step number for storing the spacecraft-to-planet form factors
OINC	- Orbit inclination
ORNT	- Transformation matrix from the vehicle coordinate system to the central coordinate system
PALB	- Planet albedo (solar reflectivity)
PERIOD	- Orbital period
PLCL	- Clock angle-to-planet position vector (in the CCS)
PLTYPE	- Spacecraft-planet form factor read/store flag
PNAME	- Name of planet being orbited

PRAD	- Planet radius
PSD	- Planet-sun distance (set to 1.0E + 15 for planetary orbits)
RATE	- Spin rate
ROTX	- Rotation about the X-axis from the VCS into the CCS
ROTY	- Rotation about the Y-axis from the VCS into the CCS
ROTZ	- Rotation about the Z-axis from the VCS into the CCS
RSUN	- Radius of sun
RTHET	- Geocentric altitude
SHADIN	- True anomaly when entering the planet shadow
SHAOUT	- True anomaly when leaving the planet shadow
SOL	- Solar "constant"
SOLO	- Extra - atmospheric solar constant
SPINT	- Transformation matrix from the CCS at zero spin time to the CCS at the current spin time
STRRA	- Right ascension of star
STRDEC	- Declination of star
SUNCO	- Cone angle to sun vector (in the CCS)
SUNCL	- Clock angle to sun vector (in the CCS)
SUNDEC	- Declination of the sun
SUNPVO	- Sun vector at noon
SUNRA	- Right ascension of the sun
STRDEC	- Declination of star
STRRA	- Right ascension of star
TIMEPR	- Current problem time
TIMEST	- Time of periapsis passage
TIMSP	- Time at which spinning begins
TRUEAN	- Orbit true anomaly
WDS	- Planet's darkside emissive power
WSS	- Subsolar emissive power of the planet
WSUN	- Emissive power of the sun



# LABELED COMMON /PLANET/

This common block contains variables used to compute the planet/ spacecraft geometry relationship and the planet element breakdown.

ALB     - Product of the planet radius and BETPM  
ALG     = AREAPL/ALB  
APEAPL - Area of the portion of the planet visible from the spacecraft  
BETPM   - Angular measurement from the subspacecraft point on the planet to the planet horizon, with the center of the planet as the vertex  
E       =  $(PRAD + H) * PRAD$   
          where:  
          PRAD = Planet radius  
          H    = Altitude of spacecraft  
POSP     - Components of a vector from the CCS origin to the center of the planet  
RADP     - Radius of a preshadowing sphere associated with a planet element  
RADPJ    - Array containing radii of the preshadowing spheres associated with each planet element

#### LABELED COMMON /PLOT/

This common block contains arrays of information used in the NPPROG and OP<sup>o</sup>PROG segments.

- IOPNNP - An array that is functionally analogous to NPNNP for use in the OPPROG segment
- IOPNV - Not used
- IOPTIT - An array that is functionally analogous to NPTIT for use in the OPPROG segment
- IOPVU - An array of views to be plotted in the OPPROG segment
- NPNNP - An array of starting locations in labeled common ARRAYS for user-input arrays of selected nodes to be plotted in the NPPROG segment
- NPTIT - An array of starting locations in labeled common ARRAYS for user-input title arrays that are to be written on plot frames in the NPPROG segment
- NPVU - An array of views to be plotted in the NPPROG segment
- OPROT - An array of user-input Euler angles for general views in the OPPROG segment
- OPRPLN - An array of the desired plot sizes of the planet radius for use in the OPPROG segment
- OPSCl - An array specifying the desired plot size for maximum spacecraft dimensions, as measured from the CCS origin for use in the OPPROG segment
- OPSClR - An array specifying the desired plot sizes of the orbit radius for use in the OPPROG segment
- OPTIMP - An array of present times in the orbit used in conjunction with OPTIMS to calculate true anomalies in the OPPROG segment
- OPTIMS - An array of perigee passage times for use in the OPPROG segment
- OPTRUE - An array of true anomalies for use in the OPPROG segment
- ZNPROT - An array of user-input Euler angles for general views in the NPPROG segment
- ZNPSCL - An array of scale factors for use in the NPPROG segment

#### LABELED COMMON /PLOTTR/

This labeled common contains the variables, flags, and Hollerith title data for controlling data plot operations.

- IPLNA - Array of node numbers for selective data plotting
- IPLSN - Array of step numbers for selective data plotting
- IPLUNT - Plot flag for type of data
- PLCRVF - Yes-no flag for curve fitting data plots
- PLLABX - X-axis plot label array
- PLLABY - Y-axis plot label array
- PLTIT1 - Plot label title 1
- PLTIT2 - Plot label title 2
- PLXMPF - Plot multiplying factor for X-axis
- PLYMPF - Plot multiplying factor for Y-axis

LABELED COMMON /PR/

A 2xNS array containing the IR and solar transmissivities associated with each surface.

PR (1, N) - IR Transmissivity of surface N

PR (2, N) - Solar transmissivity of surface N

LABELED COMMON /PSH/

A 4xNS array containing data associated with the preshadowing sphere for each surface.

PSH (1, N) - Radius of the preshadowing sphere for surface N

PSH (2, N) -

PSH (3, N) - } Components of the position vector to the center of  
PSH (4, N) - } the preshadowing sphere for surface N

LABELED COMMON /QAP/

This labeled common contains an array of absorbed planetary heat rates for each node.

QAP (N) - Absorbed planetary heat rate (energy/unit time) for  
node N

LABELED COMMON /QAR/

This labeled common contains an array of absorbed reflected (planetary albedo) heat rates for each node.

QAR (N) - Absorbed albedo heat rate (energy/unit time) for node N

LABELED COMMON /QAS/

This labeled common contains an array of absorbed solar heat rates for each node.

QAS (N) - Absorbed solar heat rate (energy/unit time) for node N

LABELED COMMON /QAVERG/

This labeled common contains an array of orbital average absorbed total heat rates for each node.

QAVERG (N) - Average absorbed heat rate for node N

LABELED COMMON /QDP/

This labeled common contains an array of direct planetary heat fluxes for each node.

QDP (N) - Direct planetary heat flux (energy/unit time/unit area)  
for node N

LABELED COMMON /QDR/

This labeled common contains an array of reflected (planetary albedo heat fluxes for each node.

LABELED COMMON /QDS/

This labeled common contains an array of solar heat fluxes for each node.

LABELED COMMON /QUCMB/

This labeled common contains an array used for working storage of correspondence data when combining absorbed-Q data for output.

- ICOMB - a. An array of node numbers identical with the list on the right side of an equal sign in correspondence data  
b. On the second pass of the combining logic, node numbers duplicated in the correspondence data are set negative

LABELED COMMON /QOFRST/

This labeled common contains an array of pointers that refer to node numbers duplicated in the correspondence data.

IFRST - An array of sequence numbers pointing to node numbers on the left side of equal signs in the correspondence data that are duplicated by node numbers on the right of any equal sign

LABELFD COMMON /QONODT/

This labeled common contains a temporary-node-number array.

NODET - A temporary-node-number array used in node combining operations. Combine operations begin in a manner identical to those in the NODE (uncombined) array and end as a combined node array

LABELED COMMON /RBDAT1/

This common block serves the same purpose in the RBPROG segment that FFDAT1 serves in the FFPROG segment (see the FFDAT1 description).

LABELED COMMON /RBFFVI/

This common block contains a row of IR area-form factor products that were read in from the NSCR1 file or the NFFR file.

FFVALI - IR area-form factor products

LABELED COMMON /RBFFVS/

This common block contains a row of solar area-form factor products that were read in from the NSCR1 file or the NFFR file.

FFVALS - Solar area-form factor products

LABELED COMMON /RBSHDC/

This common block serves the same purpose in the RBPROG segment as DRSHDC serves in the DRPROG segment (see the DRSHDC description).

LABELED COMMONS /RBSUR1/ and /RBSUR2/

These common blocks serve the same purpose in the RBPROG segment as DISUR1 serves in the DIPROG segment (see the DISUR1 description).

LABELED COMMON /RBVALI/

This common block contains a row of IR area-image factor products.

RBVALI - IR area-image factor products

LABELED COMMON /RBVALS/

This common block contains a row of solar area-image factor products.

RBVALS - Solar area-image factor products

LABELED COMMON /RKCMB/

This common block serves the same purpose in the RKPROG segment as QOCMB serves in the QOPROG segment (see the QOCMB description).

LABELED COMMON /RKFRST/

This common block serves the same purpose in the RKPROG segment as QOFRST serves in the QOPROG segment (see the QOFRST description).

#### LABELED COMMON /RKNODT/

This common block serves the same purpose in the RKPROG segment as QONODT serves in the QOPROG segment (see the QONODT description).

#### LABELED COMMONS /RMASS1/ and /RMASS2/

These common blocks serve the same purpose in the FFPROG segment as DISUR1 serves in the DIPROG segment (see the DISUR1 description).

#### LABELED COMMON /RMASSM/

This common block serves the same purpose in the RBPROG segment as DISUR1 serves in the DIPROG segment (see the DISUR1 description).

# LABELED COMMON /RNDMRP/

This common block contains the random-access record numbers and record lengths that are passed from the preprocessor to the processor to enable the processor to access preprocessor-generated data from the random access file. These variables are used in the RDPROG segment to set up processor common blocks.

NLAD	- The record length of the user array directory array
NLRIO	- The maximum number of records that can exist on the random I/O file
NRAND	- The random I/O record number of the user array name directory array
NRAPD	- The random I/O record number of the user array position directory array
NRAV, NLAV	- The random I/O record number and the record length of the user array value array. Used to set up common ARRAYS
NRBCSD, NLBCSD	- The random I/O record number and the record length of the BCS directory
NRBCSN, NLBCSN	- The random I/O record number and the record length of the BCS name array. Used to set up common BCSN
NRBCSR, NLBCSR	- The random I/O record number and the record length of the BCS index. Used to set up common BLKDIR
NRCQD, NLCQD	- The random I/O record number and the record length of the control constants directory array
NRCQV, NLCQV	- The random I/O record number and the record length of the control constants value array. Used to set up common CCONST
NRIN, NLIN	- The random I/O record number and the record length of the node index
NRIS, NLIS	- The random I/O record number and the record length of the surface index
NRSD, NLSD	- The random I/O record number and the record length of the step directory. Used to set up common ISTEP
NRT, NLT	- The random I/O record number and the record length for the title array. Used to set up common TITLE
NRTD, NLTD	- The random I/O record number and the record length of the combined directory
NRUQD, NLUQD	- The random I/O record number and the record length of the user constant directory array
NRUQV, NLUQV	- The random I/O record number and the record length of the user constant value array. Used to set up common UCONST



LABELED COMMON /SF/

This labeled common contains a storage array for script-F (gray-body factor) data.

SF - An array used to store one row of script-F values during node-combining operations in the RKPROG segment

LABELED COMMON /SFS/

This labeled common contains a storage array for script-F data.

SFS - An array used to store one row of script-F values during node-combining operations in the RCPROG segment

LABELED COMMON /SFSHDC/

This labeled common used in the SFPROG segment in same manner as labeled common DRSHDC in DRPROG.

LABELED COMMON /SFQDP/

Not used

LABELED COMMON /SFQDR/

Not used

LABELED COMMON /SFQDS/

Not used

LABELED COMMON /SFSUR1/

This labeled common contains the same variable names and is used identically with labeled common DISUR1.

#### LABELED COMMON /SFVECC/

This labeled common contains variables used to define surface and planet elements used in shadow-factor tape calculations.

NEPT - Number of elements on the planet  
NEST - Number of elements on the node  
PLAV - Planet-element area vector array (3 components in the CCS  
for each element)  
PLPV - Planet-element position vector array  
SFAV - Surface-element area vector array  
SFPV - Surface-element position vector array  
SUNPV - Sun position vector

#### LABELED COMMON /SPACE/

This labeled common contains an array of script-F (gray-body) factors from each node to space.

#### LABELED COMMON /SPACNO/

This labeled common contains radiation conductor values from each *combined* node to space, and is used in the RKPROG segment.

#### LABELED COMMON /SPCNO/

This labeled common, used in RCPRG, is exactly analogous to SPACNO.

#### LABELED COMMON /SREFLI/

This common block contains an array of surface IR specular reflectivities used in the DRPROG and RBPROG segments.

SREFLI - Array of IR specular reflectivities for active specular surfaces

LABELED COMMON /SREFLS/

This common block contains an array of surface solar specular reflectivities used in the DRPROG and RBPROG segments.

SREFLS - Array of solar specular reflectivities for active specular surfaces

LABELED COMMON /SRIR/

This common block contains an array of nodal IR specular reflectivities used in the DRPROG and the RBPROG segments.

SRIR - Array of IR specular reflectivities for active nodes

LABELED COMMON /SRSO/

This common block contains an array of nodal solar specular reflectivities used in the DRPROG and the RBPROG segments.

SRSO - Array of solar specular reflectivities for active nodes

LABELED COMMON /SUN/

This labeled common is identical to labeled common DRSUN.

LABELED COMMON /TAPE/

This labeled common contains all variable names for the processor library files (see Section III-C).

# LABELED COMMON /TITLE/

This labeled common contains title information.

TITLE	- Problem title input by the user
NTITLE	- Title identifying the segment in the segment header printouts
DTE	- Sun data
IHSTEP	- Step number in hollerith format
IPAGE	- Page number counter
LINE	- Line number counter
MLINE	- Maximum lines allowed per page
MODELN	- Configuration name
NJOB	- Job identification number
TME	- Time of day

#### LABELED COMMON /TRANS/

This labeled common contains a transformation matrix to transform planet-oriented vectors to the user-defined VCS.

PLDC - Transformation matrix used to transform vectors from a planet-centered coordinate system to the VCS

#### LABELED COMMON /TRASH/

This labeled common contains miscellaneous vector arrays and variables used in direct flux calculations.

IN - Current-node sequence number  
NCHECK - Check flag to eliminate unnecessary calls to the pre-shadowing routine  
NELT - Temporary storage address for the number of elements on a node during element optimization operations  
NSHAD - Number of possible shadowing surfaces associated with a node  
NSHADR - Counter used in conjunction with NCHECK to eliminate unnecessary calls to pre-shadowing routine  
PLAVT - Temporary storage address for planet-element area vectors  
PLPVT - Temporary storage address for planet-element position vectors  
SFAVT - Temporary storage address for node-element area vectors  
SFPVT - Temporary storage address for node-element position vectors  
SUNFVT - Temporary storage address for the sun position vector

#### LABELED COMMON /TRIR/

This common block contains an array of nodal IR transmissivities.

TRIR - Array of IR transmissivities for active nodes

#### LABELED COMMON /TRSO/

This common block contains an array of nodal solar specular transmissivities.

TRSO - Array of solar transmissivities for active nodes

LABELED COMMON /TSTR/

This common block contains an array of transformation matrices for shadowing surfaces.

TSTR - Array of transformation matrices relating the SCS coordinates of shadowing surfaces to the CCS

LABELED COMMON /UCONST/

This array contains user-defined constants that were input in the Quantities Data block.

LABELED COMMON /VARBL/

This common block contains variables used to calculate albedo and planetary fluxes in the DIPROG segment.

ALBF - Product of the solar constant and form factor from an isolated planetary element to the sun

PLNF - Emissive power of a planetary element

LABELED COMMON /VECTOR/

This labeled common contains the same variable names, used in the same way as those in labeled common SFVECC.

LABELED COMMON /XSPACE/

This labeled common contains a scratch array used to store one row of the inverted gray-body factor matrix.

BLANK COMMON

Blank common in the TRASYS processor is used as a scratch storage area in central memory.

E. ENTRY POINTS PROCESSOR LIBRARY

<u>ENTRY POINT</u>	<u>SUBROUTINE</u>
ADD	BUILDG
ADDS	BUILDG
DITTPS	DITTP
FFPNTR	FFPRNT
AQINIT	INITF
DLINIT	INITF
FFINIT	INITF
GBINIT	INITF
QOINIT	INITF
RSTOFF	RESTRT
RSTON	RESTRT
SFLSHA	SFPAKA
SFLSHE	SFPAKE
DINITS	SINITF
FFNITS	SINITF
GBNITS	SINITF
NDATAS	NDATA
ODATAS	ODATA
READMS	RIOPAC
WRITMS	

RECORDING PAGE BLANK NOT FILMED